Arm® Cortex®-M0

32-Bit Microcontroller

NuMicro® Family
Nano102/112 Series
Technical Reference Manual

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1 GENERAL DESCRIPTION

The Nano112 series ultra-low-power 32-bit microcontroller embedded with ARM® Cortex®-M0 core operates at low voltage range from 1.8V to 3.6V and runs up to 32 MHz frequency with 16/32 Kbytes embedded Flash and 4/8 Kbytes embedded SRAM and 4 Kbytes Flash loader memory for In-System Programming (ISP). The Nano112 series integrates 4 COM x 36 SEG or 6 COM x 34 SEG LCD controller, RTC, 12-bit SAR ADC, comparators and provides high performance connectivity peripheral interfaces such as UART, SPI, I^{2}C, GPIOs, and ISO-7816-3 for Smart card. The Nano112 series supports Brown-out Detector, Power-down mode with RTC turn on, RAM retention is less than 1.5 μA, Deep Power-down mode with RAM retention is less than 650 nA and fast wake-up via many peripheral interfaces.

The Nano112 series provides low voltage, low operating power consumption, low standby current, high integration peripherals, high-efficiency operation, fast wake-up function and the lowest cost 32-bit microcontrollers. The Nano112 series is suitable for a wide range of battery device applications such as:

- Wearable Device
- Smart Watch
- Wireless Gaming Control
- Hand-Held Medical Device
- RFID Reader
- Mobile Payment Smart Card Reader
- Security Alarm System
- Smart Home Appliance
- Wireless Thermostats
- Wireless Sensors Node Device (WSND)
- Wireless Auto Meter Reading (AMR)
- Portable Wireless Data Collector
- Smart Water, Gas, Heat Meters

The Nano112 series includes two product lines: Nano102 Base line and Nano112 LCD line.

The Nano102 Base line, an ultra-low-power 32-bit microcontroller embedded with ARM® Cortex®-M0 core, operates at low voltage range from 1.8V to 3.6V and runs up to 32 MHz frequency with 16/32 Kbytes embedded flash and 4/8 Kbytes embedded SRAM and 4 Kbytes Flash loader memory for In-System Programming (ISP). It integrates RTC, 8- channels 12-bit SAR ADC, 2xComparators and provides high performance connectivity peripheral interfaces such as 2 x Low Power UARTs, 2 x SPIs, 2 x I^{2}Cs, GPIOs, and 2 x ISO-7816-3 for Smart card. The Nano102 Base line supports Brown-out Detector, Power-down mode with RAM retention and fast wake-up via many peripheral interfaces.

The Nano112 LCD line, an ultra-low-power 32-bit microcontroller embedded with ARM® Cortex®-M0 core, operates at low voltage range from 1.8V to 3.6V and runs up to 32 MHz frequency with 16/32 Kbytes embedded flash and 4/8 Kbytes embedded SRAM and 4 Kbytes Flash loader memory for In-System Programming (ISP). It integrates 4 COM x 36 SEG or 6 COM x 34 SEG LCD controller, RTC, 8-channels 12-bit SAR ADC, 2 x Comparators and provides high performance connectivity peripheral interfaces such as 2 x Low Power UARTs, 2 x SPIs, 2 x I^{2}Cs, GPIOs, and 2 x ISO-7816-3 for Smart card. The Nano112 LCD line supports Brown-out Detector, Power-down mode with RAM retention and fast wake-up via many peripheral interfaces.

Product Line	UART	SPI	I^{2}C	ADC	ACMP	RTC	SC	Timer	LCD
Nano102	●	●	●	●	●	●	●	●	
Nano112	●	●	●	●	●	●	●	●	●

Table 1‑1 Connectivity Support Table

2 FEATURES

The equipped features are dependent on the product line and their sub products.

2.1 Nano102/Nano112 Features

- Low Supply Voltage Range: 1.8 V to 3.6 V
- Operating Temperature: -40°C~85°C
- Ultra-Low Power Consumption
  - Normal mode: 142 μA/MHz
  - Power-down mode with RTC on and RAM retention: 1.5 μA
  - Power-down mode and RAM retention: 650 nA
- Three power modes
  - Normal mode
  - Idle mode
  - Power-down mode
- Wake-up sources
  - RTC, WDT, I^{2}C, Timer, UART, SPI, BOD, GPIO
- Fast wake-up from Power-down mode: less than 6 μs
- Brown-out
  - Built-in 2.5V/2.0V/1.7V BOD for wide operating voltage range operation
- One built-in temperature sensor with 1℃ resolution
- Core
  - ARM® Cortex®-M0 core running up to 32 MHz
  - One 24-bit system timer
  - Supports Low Power Sleep mode
  - Single-cycle 32-bit hardware multiplier
  - NVIC for the 32 interrupt inputs, each with 4-levels of priority
  - Serial Wire Debug supports with 2 watchpoints/4 breakpoints
- Flash EPROM Memory
  - 16/32 Kbytes application program memory (APROM)
  - 4 KB in system programming (ISP) loader program memory (LDROM)
  - Programmable data flash start address and memory size with 512 bytes page erase unit
  - In System Program (ISP)/In Application Program (IAP) to update on-chip Flash EPROM
- SRAM Memory
  - 4/8 Kbytes embedded SRAM
  - Supports DMA mode
- DMA: Supports Five channels including four PDMA channels and one CRC channel
  - PDMA
    - Three modes: peripheral-to-memory, memory-to-peripheral, and memory-to-memory

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

transfer

- ■ Source address and destination address must be word alignment in all modes.
- ■ Memory-to-memory mode: transfer length must be word alignment.
- ■ Peripheral-to-memory and memory-to-peripheral mode: transfer length could be word/half-word/byte alignment.
- ■ Peripheral-to-memory and memory-to-peripheral mode: transfer data width could be word/half-word/byte alignment
- ■ Supports source and destination address direction: increment, fixed, and wrap around

- ◆ CRC
  - ■ Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and CRC-32
    - ◆ CRC-CCITT: $X^{16} + X^{12} + X^{5} + 1$
    - ◆ CRC-8: $X^{8} + X^{2} + X + 1$
    - ◆ CRC-16: $X^{16} + X^{15} + X^{2} + 1$
    - ◆ CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$

- Clock Control
  - ◆ Build-in 12/16 MHz OSC (HIRC) has 2 % deviation within all temperature range. Deviation could be reduced to 1% if turning on auto-trim function. Supports one PLL, up to 32 MHz, for high performance system operation
  - ◆ External 4~24 MHz(HXT) crystal input for precise timing operation
  - ◆ Low power 10 kHz OSC(LIRC) for watchdog and low power system operation
  - ◆ External 32.768 kHz(LXT) crystal input for RTC and low power system operation
- GPIO
  - ◆ Three I/O modes:
    - ■ Push-Pull output
    - ■ Open-Drain output
    - ■ Input only with high impendence
  - ◆ All inputs with Schmitt trigger
  - ◆ I/O pin configured as interrupt source with edge/level setting
  - ◆ Supports input 5V tolerance, except
    - ■ PA.0 ~ PA.7 (sharing pin with ADC),
    - ■ PA.12~ PA.13 (sharing pin with comparator),
    - ■ PF.0 ~ PF.1 (sharing pin with LXT).
- Timer
  - ◆ Supports 4 sets of 32-bit timers, each timer with 24-bit up-counting timer and one 8-bit pre-scale counter
  - ◆ Each timer could have independent clock source selection
  - ◆ Supports one-shot, periodic, output toggle and continuous operation modes
  - ◆ Internal trigger event to ADC and PDMA

  - Supports PDMA mode
  - Wake system up from Power-down mode
- Watchdog Timer
  - Clock Source from LIRC (Internal 10 kHz Low Speed Oscillator Clock)
  - Selectable time-out period from 1.6 ms ~ 26 sec (depending on clock source)
  - Interrupt or reset selectable when watchdog time-out
  - Wakes system up from Power-down mode
- Window Watchdog Timer(WWDT)
  - 6-bit down counter and 6-bit compare value to make the window period flexible
  - Selectable WWDT clock pre-scale counter to make WWDT time-out interval variable.
- RTC
  - Supports software compensation by setting frequency compensate register (FCR)
  - Supports RTC counter (second, minute, hour) and calendar counter (day, month, year)
  - Supports Alarm registers (second, minute, hour, day, month, year)
  - Selectable 12-hour or 24-hour mode
  - Automatic leap year recognition
  - Supports periodic time tick interrupt with 8 periodic options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second
  - Wake system up from Power-down mode
  - Supports 80 bytes spare registers and a snoop pin to clear the content of these spare registers
  - Supports 1, 1/2, 1/4, 1/8, 1/16 Hz clock output
- PWM/Capture
  - Supports 1 PWM module with two 16-bit PWM generators
  - Provides four PWM outputs or two complementary paired PWM outputs
  - Each PWM generator equipped with one clock divider, one 8-bit prescaler, two clock selectors, and one Dead-zone generator for complementary paired PWM
  - (Shared with PWM timers) with four 16-bit digital capture timers provides four rising/ falling/both capture inputs.
  - Supports One-shot and Continuous mode
  - Supports Capture interrupt
- UART
  - Up to 1 Mbit/s baud rate and support 9600 baud rate at 32.768 kHz
  - Up to two 16-byte FIFO UART controllers
  - UART ports with flow control (TX, RX, CTSn and RTSn)
  - Supports IrDA (SIR) function
  - Supports LIN function
  - Supports RS-485 9 bit mode and direction control.
  - Programmable baud rate generator

  - Supports PDMA mode
  - Wakes system (CTSn, received data or RS-485 address matched) up from Power-down mode
- SPI
  - Up to two sets of SPI controllers
  - Supports Master (max. 32 MHz) or Slave (max. 16 MHz) mode operation
  - Supports 1 bit and 2 bit transfer mode
  - Support Dual IO transfer mode
  - Configurable bit length of a transaction from 8 to 32-bit
  - Supports MSB first or LSB first transfer sequence
  - Two slave select lines supported in Master mode
  - Configurable byte or word suspend mode
  - Supports byte re-ordering function
  - Supports variable serial clock in Master mode
  - Provide separate 8-level depth transmit and receive FIFO buffer
  - Supports wake-up function(SPI clock toggle in Power-down mode)
  - Supports PDMA transfer
  - Supports 3-wires, no slave select signal, bi-direction interface
- I^{2}C
  - Up to two sets of I^{2}C devices
  - Master/Slave up to 1 Mbit/s
  - Bi-directional data transfer between masters and slaves
  - Multi-master bus (no central master)
  - Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
  - Serial clock synchronization allows devices with different bit rates to communicate via one serial bus
  - Serial clock synchronization used as a handshake mechanism to suspend and resume serial transfer
  - Built-in 14-bit time-out counter requesting the I^{2}C interrupt if the I^{2}C bus hangs up and timer-out counter overflows
  - Programmable clocks allowing for versatile rate control
  - Supports 7-bit addressing mode
  - Supports multiple address recognition (four slave addresses with mask option)
  - Wake system up(address match) from Power-down mode
- ADC
  - 12-bit SAR ADC up to 1.6MSPS conversion rate
  - Up to 12 channels: 8 external channel(PA.0 ~ PA.7) and 4 internal channels.
  - Four internal channels: internal reference voltage (Int_V$_{REF}$), Temperature sensor, AV$_{DD}$, and

  AV_{SS}.
  - Supports three reference voltage sources: V_{REF} pin, internal reference voltage (Int_V_{REF}), and AV_{DD}.
  - Supports Single Scan, Single Cycle Scan, and Continuous Scan mode
  - Each channel with individual result register
  - Threshold voltage detection (comparator function)
  - Conversion started by software programming or external input
  - Supports PDMA mode
  - Supports up to four timer time-out events (TMR0, TMR1, TMR2 and TMR3) to enable ADC
- Smart Card (SC)
  - Compliant to ISO-7816-3 T=0, T=1
  - Supports up to two ISO-7816-3 ports
  - Separates receive/transmit 4 bytes entry FIFO for data payloads
  - Programmable transmission clock frequency
  - Programmable receiver buffer trigger level
  - Programmable guard time selection (11 ETU ~ 267 ETU)
  - A 24-bit and two 8-bit time-out counters for Answer to Request (ATR) and waiting times processing
  - Supports auto inverse convention function
  - Supports transmitter and receiver error retry and error limit function
  - Supports hardware activation sequence process
  - Supports hardware warm reset sequence process
  - Supports hardware deactivation sequence process
  - Supports hardware auto deactivation sequence when detect the card is removal
  - Supports UART mode (full–duplex)
- ACMP
  - Supports up to 2 analog comparators
  - Analog input voltage range: 0 ~ AV_{DD}
  - Supports Hysteresis function
  - Two analog comparators with optional internal reference voltage input at negative end
- 96-bit unique ID
- 128-bit unique customer ID
- Packages:
  - All Green package (RoHS)
  - LQFP 64-pin(7x7) / 48-pin(7x7)/ QFN33-pin(5x5)

2.2 Nano112 Features – LCD Line

- Low Supply Voltage Range: 1.8 V to 3.6 V
- Ultra-Low Power Consumption
  - Operation mode: 150 μA/MHz
  - Power-down mode: 1.5 μA (RTC on, RAM retention)
  - Deep Power-down mode: 650 nA (RAM retention)
- Fast Wake-Up From Standby Mode: Less than 6 μs
- Core
  - ARM® Cortex®-M0 core running up to 32 MHz
  - One 24-bit system timer
  - Supports Low Power Sleep mode
  - Single-cycle 32-bit hardware multiplier
  - NVIC for the 32 interrupt inputs, each with 4-levels of priority
  - Serial Wire Debug supports with 2 watchpoints/4 breakpoints
- Flash EPROM Memory
  - Runs up to 32 MHz with zero wait state for discontinuous address read access.
  - 16/32 Kbytes application program memory (APROM)
  - 4 Kbytes In System Programming (ISP) loader program memory (LDROM)
  - Programmable data flash start address and memory size with 512 bytes page erase unit
  - In System Program (ISP)/In Application Program (IAP) to update on chip Flash EPROM
- SRAM Memory
  - 4/8 Kbytes embedded SRAM
  - Supports DMA mode
- DMA: Supports 5 channels: 4 PDMA channels, and one CRC channel
  - PDMA
    - Peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfer
    - Supports word boundary address
    - Supports word alignment transfer length in memory-to-memory mode
    - Supports word/half-word/byte alignment transfer length in peripheral-to-memory and memory-to-peripheral mode
    - Supports word/half-word/byte transfer data width from/to peripheral
    - Supports address direction: increment, fixed, and wrap around
  - CRC
    - Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and CRC-32
      - CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
      - CRC-8: $X^8 + X^2 + X + 1$
      - CRC-16: $X^{16} + X^{15} + X^2 + 1$

        - CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$

- Clock Control
    - Flexible selection for different applications
    - Built-in 12/16 MHz OSC, can be trimmed to 1 % deviation within all temperature range when turning on auto-trim function (system must have external 32.768 kHz crystal input) otherwise 12/16 MHz OSC has 2 % deviation within all temperature range.
    - Low power 10 kHz OSC for watchdog and low power system operation
    - Supports one PLL, up to 32 MHz, for high performance system operation
    - External 4~24 MHz crystal input for precise timing operation
    - External 32.768 kHz crystal input for RTC function and low power system operation
- GPIO
    - Three I/O modes:
        - Push-Pull output
        - Open-Drain output
        - Input only with high impendence
    - All inputs with Schmitt trigger
    - I/O pin configured as interrupt source with edge/level setting
    - Supports High Driver and High Sink I/O mode
    - Supports input 5V tolerance, except PA.0 ~ PA.7, PA.12, PA.13, P.0(X32I), PF.1(X32O)
- Timer
    - Supports 4 sets of 32-bit timers, each with 24-bit up-timer and one 8-bit pre-scale counter
    - Independent Clock Source for each timer
    - Provides one-shot, periodic, output toggle and continuous operation modes
    - Internal trigger event to ADC and PDMA
    - Supports PDMA mode
    - Wake system up from Power-down mode
- Watchdog Timer
    - Clock Source from LIRC (Internal 10 kHz Low Speed Oscillator Clock)
    - Selectable time-out period from 1.6 ms ~ 26 sec (depending on clock source)
    - Interrupt or reset selectable when watchdog time-out
    - Wake system up from Power-down mode
- Window Watchdog Timer(WWDT)
    - 6-bit down counter and 6-bit compare value to make the window period flexible
    - Selectable WWDT clock pre-scale counter to make WWDT time-out interval variable.
- RTC
    - Supports software compensation by setting frequency compensate register (FCR)
    - Supports RTC counter (second, minute, hour) and calendar counter (day, month, year)

  - Supports Alarm registers (second, minute, hour, day, month, year)
  - Selectable 12-hour or 24-hour mode
  - Automatic leap year recognition
  - Supports periodic time tick interrupt with 8 periodic options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second
  - Wake system up from Power-down mode
  - Supports 80 bytes spare registers and a snoop pin to clear the content of these spare registers
  - Supports 1, 1/2, 1/4, 1/8, 1/16 Hz clock output
- PWM/Capture
  - Supports 1 PWM module with two 16-bit PWM generators
  - Provides four PWM outputs or two complementary paired PWM outputs
  - Each PWM generator equipped with one clock divider, one 8-bit prescaler, two clock selectors, and one Dead-zone generator for complementary paired PWM
  - (Shared with PWM timers) with four 16-bit digital capture timers provides four rising/ falling/both capture inputs.
  - Supports Capture interrupt
- UART
  - Up to 1 Mbit/s baud rate and support 9600 baud rate @ 32kHz, low power mode
  - Up to two 16-byte FIFO UART controllers
  - UART ports with flow control (TX, RX, CTSn and RTSn)
  - Supports IrDA (SIR) function
  - Supports LIN function
  - Supports RS-485 9 bit mode and direction control (Low Density Only)
  - Programmable baud rate generator
  - Supports PDMA mode
  - Wake system up (CTS, received data or RS-485 address matched) from Power-down mode
- SPI
  - Up to two sets of SPI controllers
  - Master up to 32 MHz, and Slave up to 16 MHz
  - Supports SPI/MICROWIRE Master/Slave mode
  - Full duplex synchronous serial data transfer
  - Variable length of transfer data from 4 to 32 bits
  - MSB or LSB first data transfer
  - RX and TX on both rising or falling edge of serial clock independently
  - Two slave/device select lines when SPI controller is as the master, and 1 slave/device select line when SPI controller is as the slave
  - Supports byte suspend mode in 32-bit transmission
  - Supports two channel PDMA requests, one for transmit and another for receive

  - Supports three wire mode, no slave select signal, bi-direction interface
  - Wake system up (SPI clock toggle) from Power-down mode
- I^2C
  - Up to two sets of I^2C devices
  - Master/Slave up to 1Mbit/s
  - Bidirectional data transfer between masters and slaves
  - Multi-master bus (no central master)
  - Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
  - Serial clock synchronization allowing devices with different bit rates to communicate via one serial bus
  - Serial clock synchronization used as a handshake mechanism to suspend and resume serial transfer
  - Built-in 14-bit time-out counter requesting the I^2C interrupt if the I^2C bus hangs up and timer-out counter overflows
  - Programmable clocks allow versatile rate control
  - Supports 7-bit addressing mode
  - Supports multiple address recognition (four slave address with mask option)
  - Wake system up (address match) from Power-down mode
- ADC
  - 12-bit SAR ADC up to 1Msps conversion rate
  - Up to 7-ch single-ended input from external pin (PA.0 ~ PA.6)
  - Four internal channels from internal reference voltage (Int_V_{REF}), Temperature sensor, AV_{DD}, and AV_{SS}
  - Supports three reference voltage sources from V_{REF} pin, internal reference voltage (Int_V_{REF}), and AV_{DD}.
  - Single scan/single cycle scan/continuous scan
  - Each channel with individual result register
  - Only scan on enabled channels
  - Threshold voltage detection (comparator function)
  - Conversion start by software programming or external input
  - Supports PDMA mode
  - Supports up to four timer time-out events (TMR0, TMR1, TMR2, and TMR3) to enable ADC
- Smart Card (SC)
  - Compliant to ISO-7816-3 T=0, T=1
  - Supports up to two ISO-7816-3 ports
  - Separates receive / transmit 4 bytes entry FIFO for data payloads
  - Programmable transmission clock frequency
  - Programmable receiver buffer trigger level

  - Programmable guard time selection (11 ETU ~ 267 ETU)
  - A 24-bit and two 8-bit time-out counter for Answer to Request (ATR) and waiting times processing
  - Supports auto inverse convention function
  - Supports transmitter and receiver error retry and error limit function
  - Supports hardware activation sequence process
  - Supports hardware warm reset sequence process
  - Supports hardware deactivation sequence process
  - Supports hardware auto deactivation sequence when detect the card is removal
  - Supports UART mode (full-duplex)
- ACMP
  - Supports up to 2 analog comparators
  - Analog input voltage range: 0 ~ AV_{DD}
  - Supports Hysteresis function
  - Two analog comparators with optional internal reference voltage input at negative end
- Wake-up source
  - Support RTC, WDT, I^{2}C, Timer, UART, SPI, BOD, GPIO
- LCD
  - LCD driver for up to 4 COM x 36 SEG or 6 COM x 34 SEG
  - Supports Static,1/2 bias and 1/3 bias voltage
  - Six display modes; Static, 1/2 duty, 1/3 duty, 1/4 duty, 1/5 duty and 1/6 duty.
  - Selectable LCD frequency by frequency divider
  - Configurable frame frequency
  - Internal Charge pump, adjustable contrast adjustment
  - Configurable Charge pump frequency
  - Blinking capability
  - Supports R-type/C-type/External C-type method
  - Configurable internal R-ladder resistor value (200K/300K/400K)
  - LCD frame interrupt
- One built-in temperature sensor with 1℃ resolution
- Brown-out
  - Built-in 2.5V/2.0V/1.7V BOD for wide operating voltage range operation
- 96-bit unique ID
- 128-bit unique customer ID
- Operating Temperature: -40°C~85°C
- Packages:
  - All Green package (RoHS)
  - LQFP 100-pin(14x14) / 64-pin(10x10) / 64-pin(7x7) / 48-pin(7x7)

3 ABBREVIATIONS

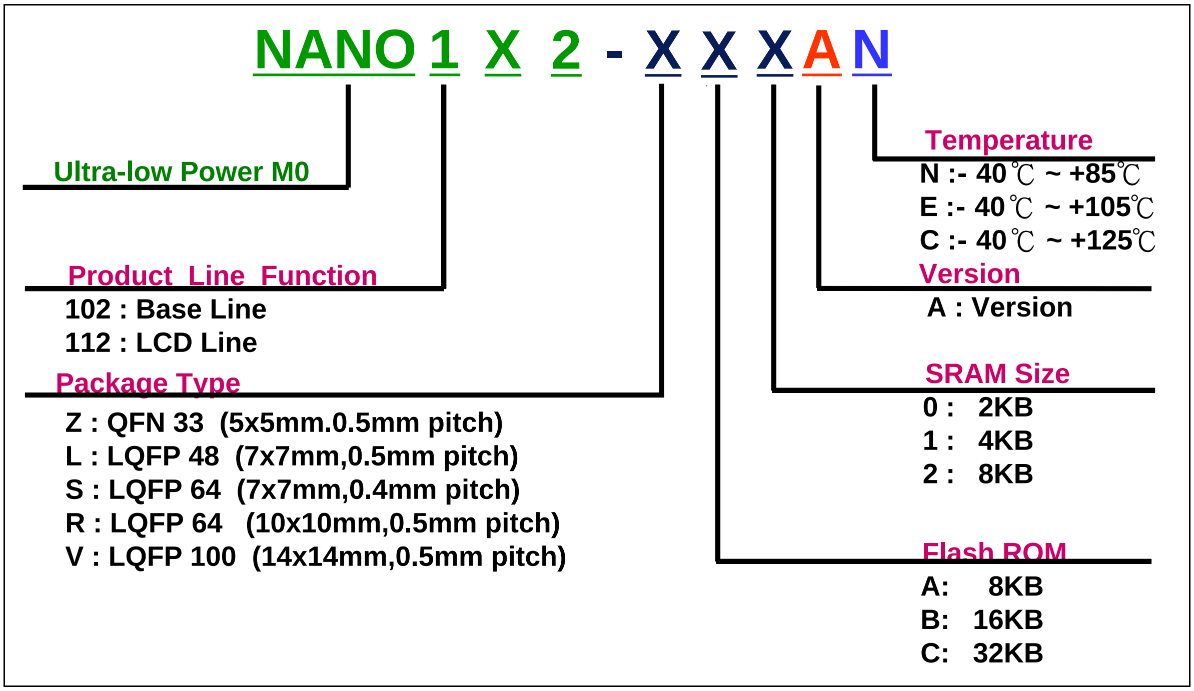
Acronym	Description
ACMP	Analog Comparator Controller
ADC	Analog-to-Digital Converter
AES	Advanced Encryption Standard
APB	Advanced Peripheral Bus
AHB	Advanced High-Performance Bus
BOD	Brown-out Detection
CAN	Controller Area Network
DAP	Debug Access Port
DES	Data Encryption Standard
EBI	External Bus Interface
EPWM	Enhanced Pulse Width Modulation
FIFO	First In, First Out
FMC	Flash Memory Controller
FPU	Floating-point Unit
GPIO	General-Purpose Input/Output
HCLK	The Clock of Advanced High-Performance Bus
HIRC	12/16 MHz Internal High Speed RC Oscillator
HXT	4~24 MHz External High Speed Crystal Oscillator
IAP	In Application Programming
ICP	In Circuit Programming
ISP	In System Programming
LDO	Low Dropout Regulator
LIN	Local Interconnect Network
LIRC	10 kHz internal low speed RC oscillator (LIRC)
MPU	Memory Protection Unit
NTC	Negative Temperature Coefficient
NVIC	Nested Vectored Interrupt Controller
PCLK	The Clock of Advanced Peripheral Bus
PDMA	Peripheral Direct Memory Access
PLL	Phase-Locked Loop
PTC	Positive Temperature Coefficient
PT1000	Thermal Resistance
PWM	Pulse Width Modulation

QEI	Quadrature Encoder Interface
SDIO	Secure Digital Input/Output
SPI	Serial Peripheral Interface
SPS	Samples per Second
TDES	Triple Data Encryption Standard
TMR	Timer Controller
UART	Universal Asynchronous Receiver/Transmitter
UCID	Unique Customer ID
USB	Universal Serial Bus
WDT	Watchdog Timer
WWDT	Window Watchdog Timer

Table 3-1 List of Abbreviations

4 PARTS INFORMATION LIST AND PIN CONFIGURATION

4.1 NuMicro® Nano102/112 Series Selection Code



NANO 1 X 2 - X X X A N

Ultra-low Power M0

Product Line Function
102 : Base Line
112 : LCD Line

Package Type
Z : QFN 33 (5x5mm.0.5mm pitch)
L : LQFP 48 (7x7mm,0.5mm pitch)
S : LQFP 64 (7x7mm,0.4mm pitch)
R : LQFP 64 (10x10mm,0.5mm pitch)
V : LQFP 100 (14x14mm,0.5mm pitch)

Temperature
N :- 40℃ ~ +85℃
E :- 40℃ ~ +105℃
C :- 40℃ ~ +125℃

Version
A : Version

SRAM Size
0 : 2KB
1 : 4KB
2 : 8KB

Flash ROM
A: 8KB
B: 16KB
C: 32KB

Figure 4-1 NuMicro® Nano112 Series Selection Code

4.2 NuMicro® Nano112 Products Selection Guide

4.2.1 NuMicro® Nano102 Base Line Selection Guide

Part Number	APROM (KB)	RAM (KB)	Data Flash (KB)	ISP ROM (KB)	I/O	Timer (32-Bit)	Connectivity			Comparator	PWM (16-Bit)	ADC (12-Bit)	RTC	PDMA	LCD	ISO-7816-3	ISP/ICP/IAP	Package
							UART	SPI	I^{2}C									
NANO102ZB1AN	16	4	Configurable	4	27	4	3	2	2	2	4	2	√	4	-	1	√	QFN33
NANO102ZC2AN	32	8	Configurable	4	27	4	3	2	2	2	4	2	√	4	-	1	√	QFN33
NANO102LB1AN	16	4	Configurable	4	40	4	4	2	2	2	4	7	√	4	-	2	√	LQFP48
NANO102LC2AN	32	8	Configurable	4	40	4	4	2	2	2	4	7	√	4	-	2	√	LQFP48
NANO102SC2AN	32	8	Configurable	4	58	4	4	2	2	2	4	7	√	4	-	2	√	LQFP64*

QFN33: 5x5 mm

LQFP48:7x7 mm

LQFP64*: 7x7 mm

4.2.2 NuMicro® Nano112 LCD Line Selection Guide

Part Number	APROM (KB)	RAM (KB)	Data Flash (KB)	ISP ROM (KB)	I/O	Timer (32-Bit)	Connectivity			Comparator	PWM (16-Bit)	ADC (12-Bit)	RTC	PDMA	ISO-7816-3	ISP/ICP/IAP	LCD	Package
							UART	SPI	I^{2}C									
NANO112LB1AN	16	4	Configurable	4	40	4	4	2	2	2	4	7	√	4	2	√	4x20, 6x18	LQFP48
NANO112LC2AN	32	8	Configurable	4	40	4	4	2	2	2	4	7	√	4	2	√	4x20, 6x18	LQFP48
NANO112SB1AN	16	4	Configurable	4	58	4	4	2	2	2	4	7	√	4	2	√	4x32, 6x30	LQFP64
NANO112SC2AN	32	8	Configurable	4	58	4	4	2	2	2	4	7	√	4	2	√	4x32, 6x30	LQFP64
NANO112RB1AN	16	4	Configurable	4	58	4	4	2	2	2	4	7	√	4	2	√	4x32, 6x30	LQFP64*
NANO112RC2AN	32	8	Configurable	4	58	4	4	2	2	2	4	7	√	4	2	√	4x32, 6x30	LQFP64*
NANO112VC2AN	32	8	Configurable	4	80	4	4	2	2	2	4	8	√	4	2	√	4x36, 6x34	LQFP100

LQFP48: 7x7 mm

LQFP64: 7x7 mm

LQFP64*: 10x10 mm

4.3 Pin Configuration

4.3.1 NuMicro® Nano102 Pin Diagrams

4.3.1.1 NuMicro® Nano102 LQFP 64-pin

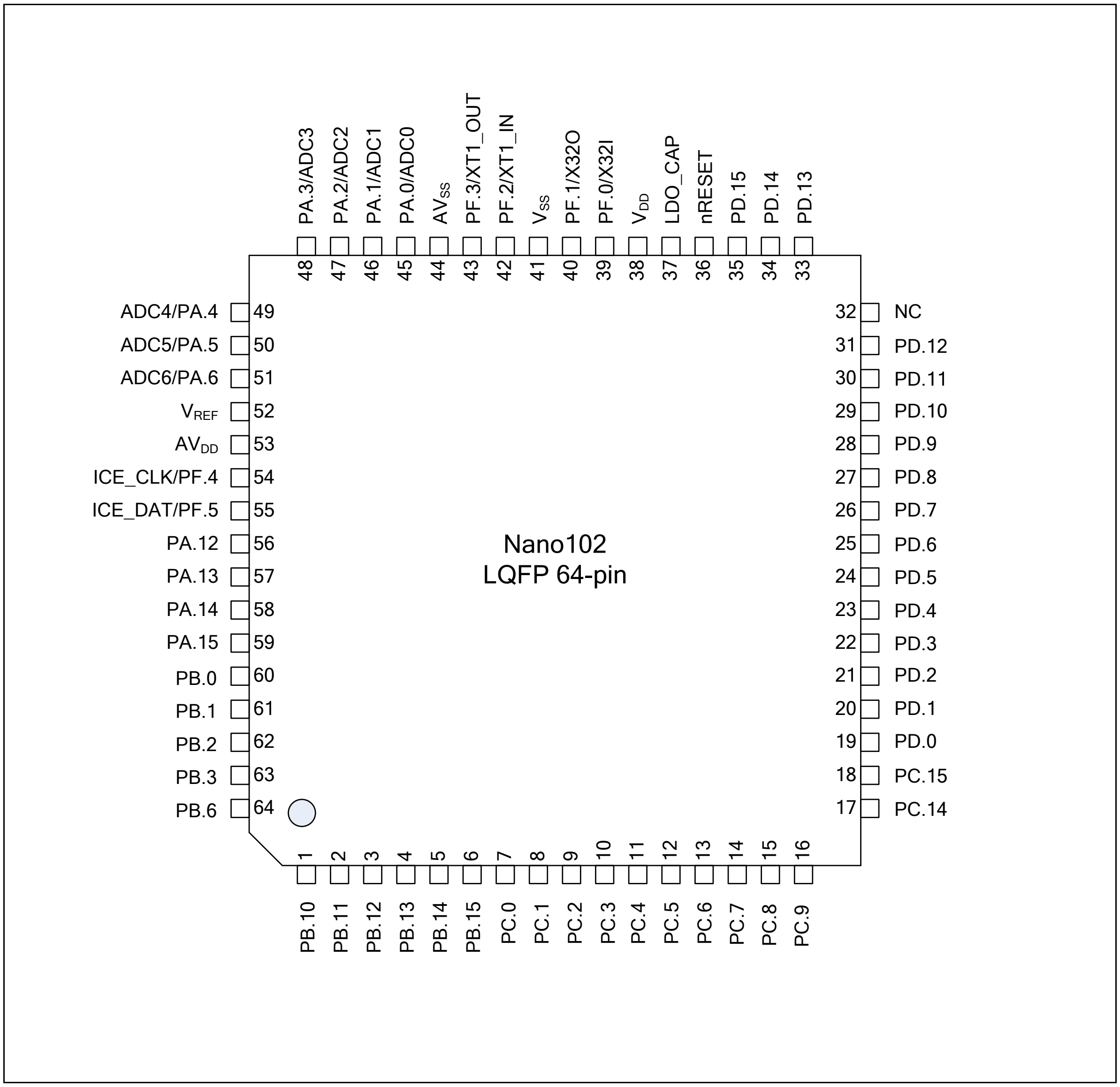


Figure 4- 2 NuMicro® Nano102 LQFP 64-pin Diagram

4.3.1.2 NuMicro® Nano102 LQFP 48-pin

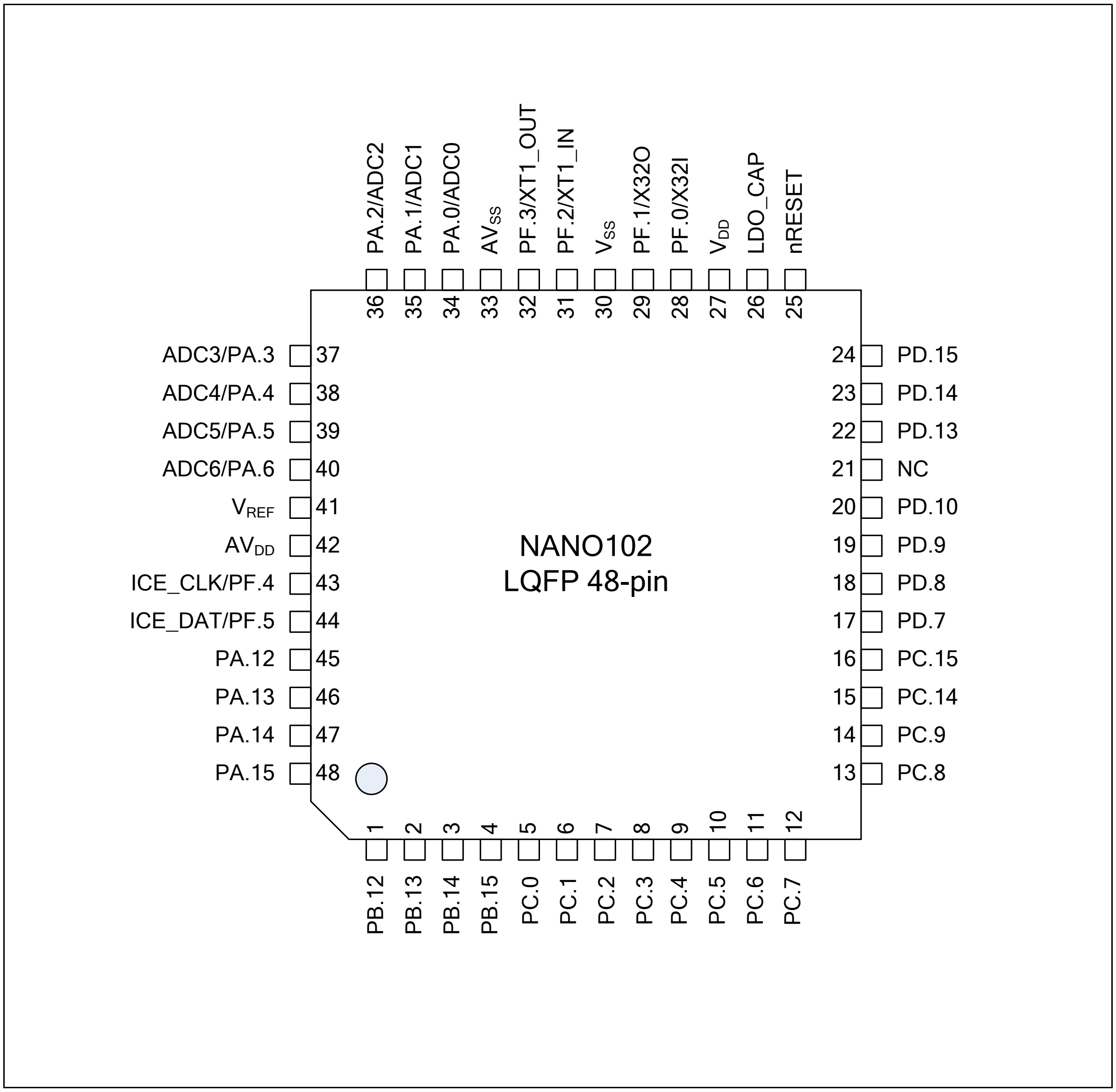


Figure 4‑3 NuMicro® Nano102 LQFP 48-pin Diagram

4.3.1.3 NuMicro® Nano102 QFN 33-pin

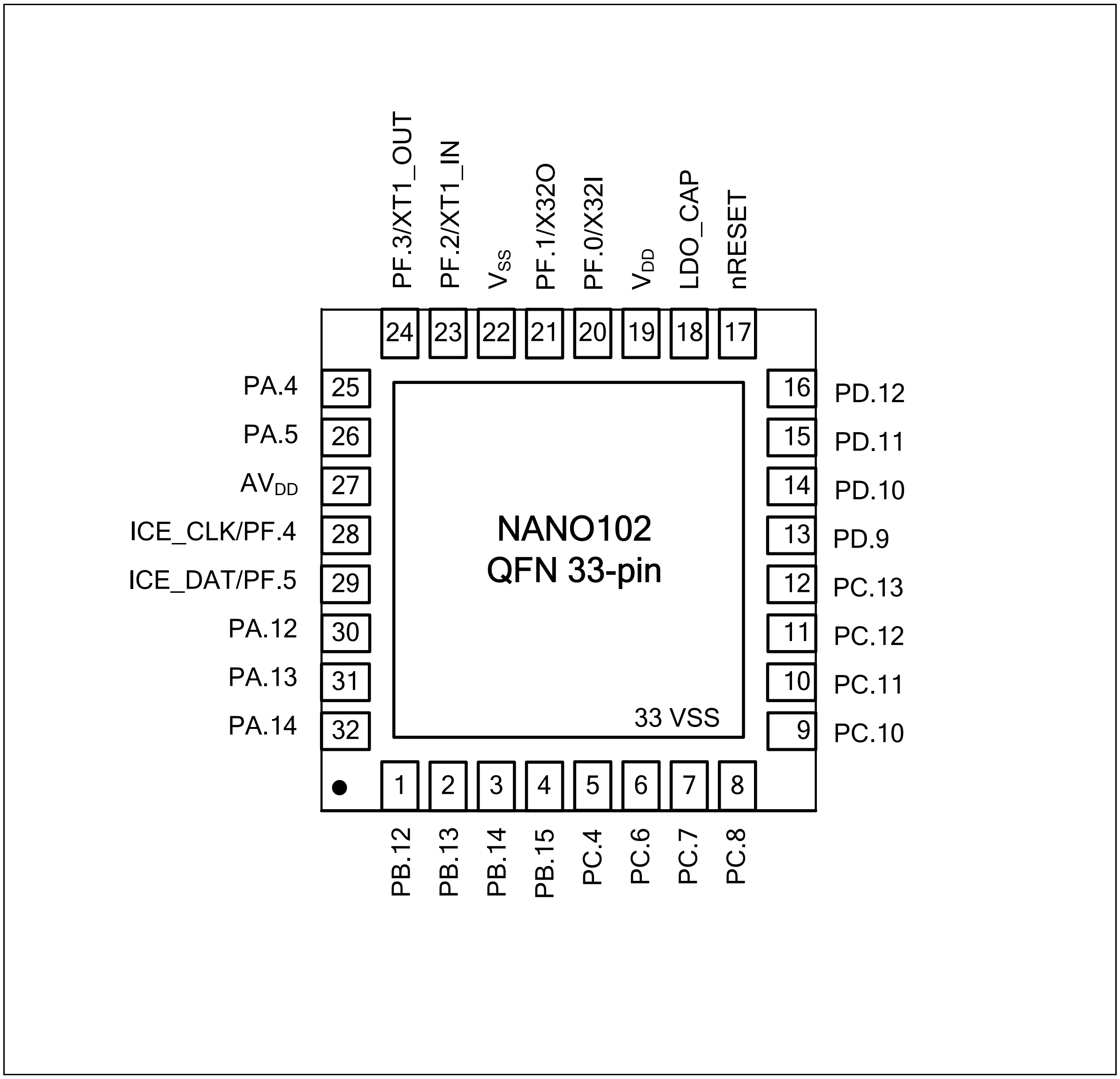


Figure 4‑4 NuMicro® Nano102 QFN 32-pin Diagram

4.3.2 NuMicro® Nano112 Pin Diagrams

4.3.2.1 NuMicro® Nano112 LQFP 100-pin

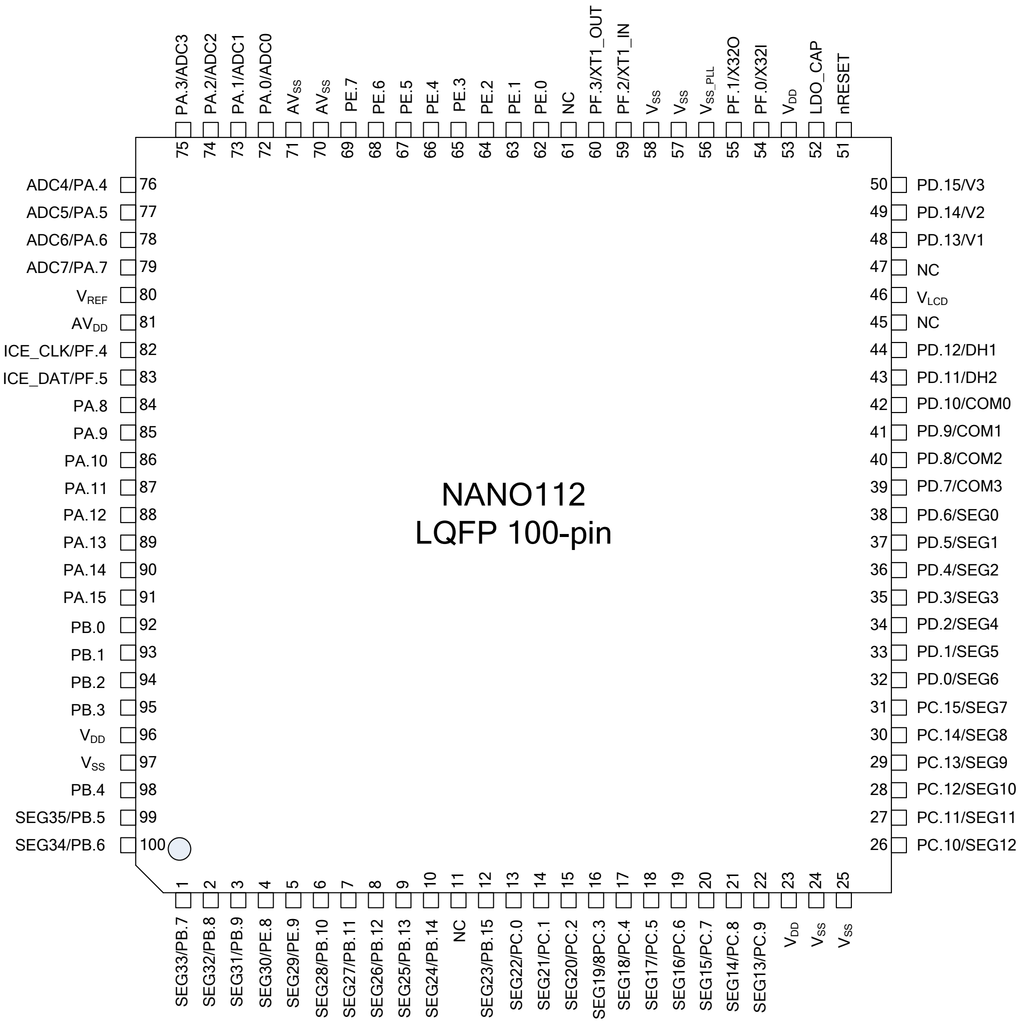


Figure 4‑5 NuMicro® Nano112 LQFP 100-pin Diagram

4.3.2.2 NuMicro® Nano112 LQFP 64-pin

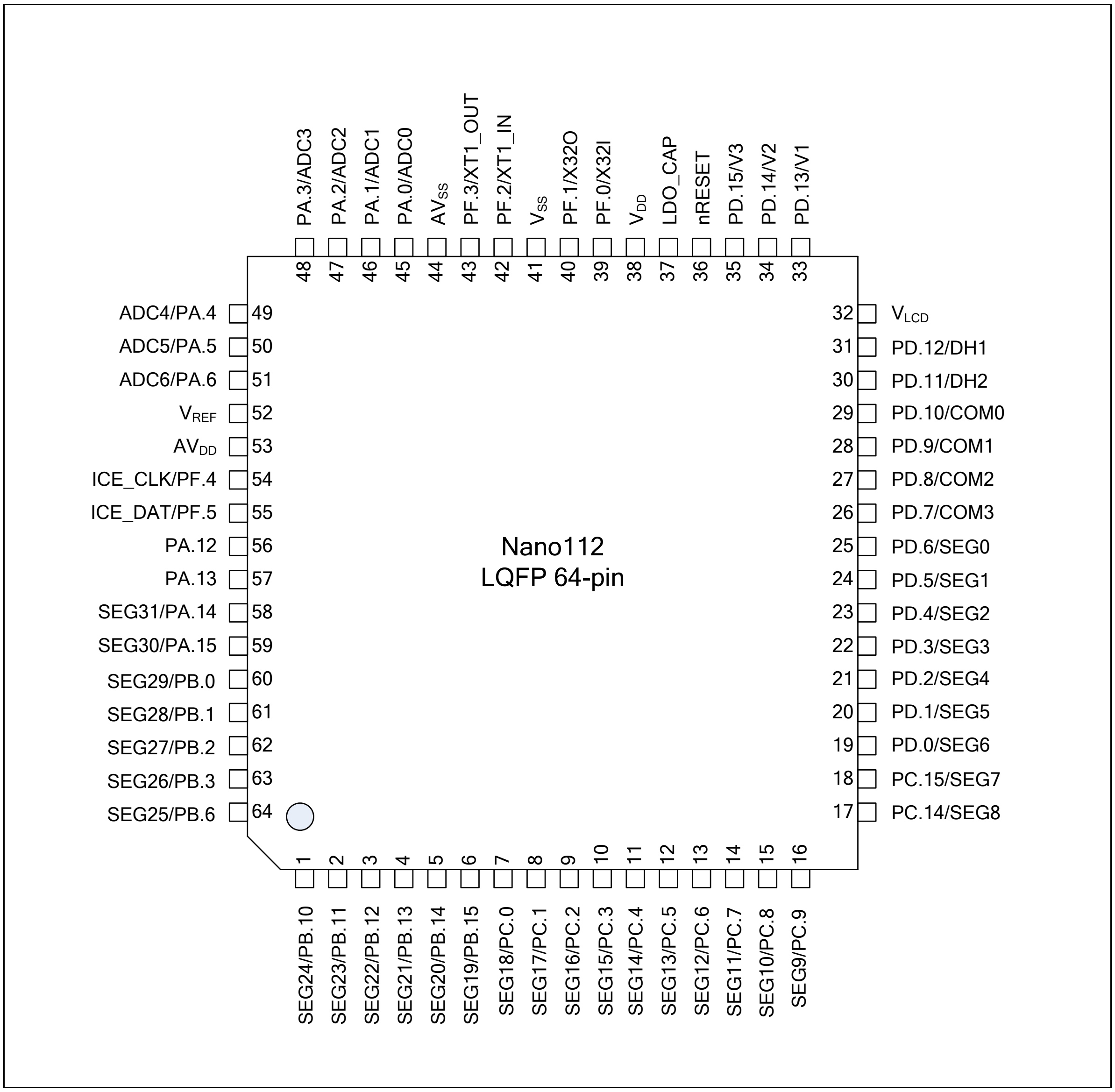


Figure 4-6 NuMicro® Nano112 LQFP 64-pin Diagram

4.3.2.3 NuMicro® Nano112 LQFP 48-pin

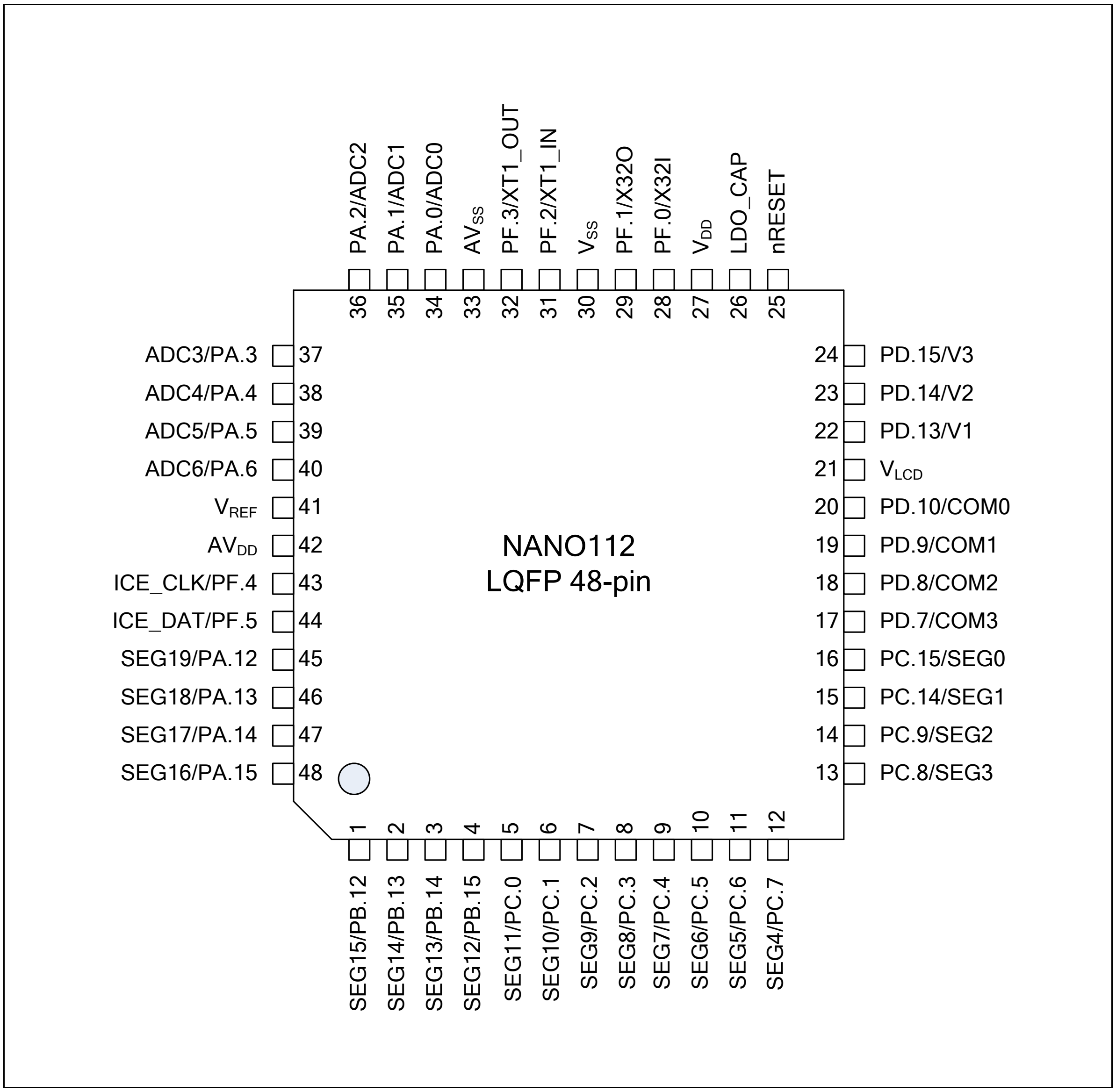


Figure 4-7 NuMicro® Nano112 LQFP 48-pin Diagram

4.4 Pin Description

4.4.1 NuMicro® Nano102 Pin Description

Pin No.			Pin Name	Pin Type	Description
64-pin	48-pin	32-pin			
1			PB.10	I/O	General purpose digital I/O pin
			UART1_RXD	I	UART1 Data receiver input pin
			SPI0_MOSI1	I/O	SPI0 2nd MOSI (Master Out, Slave In) pin
2			PB.11	I/O	General purpose digital I/O pin
			UART1_RTSn	O	UART1 Request to Send output pin
			SPI0_MISO1	I/O	SPI0 2rd MISO (Master In, Slave Out) pin
			TM1	I/O	Timer1 external counter input or Timer1 toggle out
3	1	1	PB.12	I/O	General purpose digital I/O pin
			UART0_RTSn	O	UART0 Request to Send output pin
			SPI0_MOSI0	I/O	SPI0 1st MOSI (Master Out, Slave In) pin
			TM0	I/O	Timer0 external counter input or Timer0 toggle out.
			FCLK0	O	Frequency Divider0 output pin
4	2	2	PB.13	I/O	General purpose digital I/O pin
			UART0_RXD	I	UART0 Data receiver input pin
			SPI0_MISO0	I/O	SPI0 1st MISO (Master In, Slave Out) pin
5	3	3	PB.14	I/O	General purpose digital I/O pin
			UART0_TXD	O	UART0 Data transmitter output pin (This pin could be modulated with PWM0 output.)
			SPI0_CLK	I/O	SPI0 serial clock pin
6	4	4	PB.15	I/O	General purpose digital I/O pin
			UART0_CTSn	I	UART0 Clear to Send input pin
			SPI0_SS0	I/O	SPI0 1st slave select pin
7	5		PC.0	I/O	General purpose digital I/O pin
			SPI0_SS1	I/O	SPI0 2nd slave select pin
			I2C0_SCL	I/O	I^{2}C0 clock pin
			PWM0_CH0	I/O	PWM0 Channel0 output
8	6		PC.1	I/O	General purpose digital I/O pin
			I2C0_SDA	I/O	I^{2}C0 data I/O pin
			PWM0_CH1	I/O	PWM0 Channel1 output
9	7		PC.2	I/O	General purpose digital I/O pin

Pin No.			Pin Name	Pin Type	Description
64-pin	48-pin	32-pin			
			I2C1_SCL	O	I^{2}C1 clock pin
			PWM0_CH2	I/O	PWM0 Channel2 output
10	8		PC.3	I/O	General purpose digital I/O pin
			I2C1_SDA	I/O	I^{2}C1 data I/O pin
			PWM0_CH3	I/O	PWM0 Channel3 output
11	9	5	PC.4	I/O	General purpose digital I/O pin
			UART1_CTSn	I	UART1 Clear to Send input pin
			SC0_CLK	O	SmartCard0 clock pin (SC0_UART_TXD)
			INT0	I	External interrupt0 input pin
12	10		PC.5	I/O	General purpose digital I/O pin
			SC0_CD	I	SmartCard0 card detect pin
13	11	6	PC.6	I/O	General purpose digital I/O pin
			UART1_RTSn	O	UART1 Request to Send output pin
			SC0_DAT	I/O	SmartCard0 DATA pin (SC0_UART_RXD)
14	12	7	PC.7	I/O	General purpose digital I/O pin
			UART1_RXD	I	UART1 Data receiver input pin
			SC0_PWR	O	SmartCard0 Power pin
15	13	8	PC.8	I/O	General purpose digital I/O pin
			UART1_TXD	O	UART1 Data transmitter output pin (This pin could be modulated with PWM0 output.)
			SC0_RST	O	SmartCard0 RST pin
16	14		PC.9	I/O	General purpose digital I/O pin
		9	PC.10	I/O	General purpose digital I/O pin
			I2C1_SCL	I/O	I^{2}C1 clock pin
			SC1_CD	I	SmartCard1 card detect
		10	PC.11	I/O	General purpose digital I/O pin
			I2C1_SDA	I/O	I^{2}C 1 data I/O pin
			SC1_PWR	O	SmartCard1 PWR pin
		11	PC.12	I/O	General purpose digital I/O pin
			SC1_CLK	O	SmartCard1 clock pin (SC1_UART_TXD)
		12	PC.13	I/O	General purpose digital I/O pin
			SC1_DAT	I/O	SmartCard1 DATA pin (SC1_UART_RXD)
17	15		PC.14	I/O	General purpose digital I/O pin
			SC1_CD	I	SmartCard1 card detect

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

Pin No.			Pin Name	Pin Type	Description
64-pin	48-pin	32-pin			
18	16		PC.15	I/O	General purpose digital I/O pin
			SC1_PWR	O	SmartCard1 PWR pin
19			PD.0	I/O	General purpose digital I/O pin
20			PD.1	I/O	General purpose digital I/O pin
21			PD.2	I/O	General purpose digital I/O pin
22			PD.3	I/O	General purpose digital I/O pin
23			PD.4	I/O	General purpose digital I/O pin
			SC1_RST	O	SmartCard1 RST pin
24			PD.5	I/O	General purpose digital I/O pin
25			PD.6	I/O	General purpose digital I/O pin
26	17		PD.7	I/O	General purpose digital I/O pin
			SC1_CLK	O	SmartCard1 clock pin (SC1_UART_TXD)
27	18		PD.8	I/O	General purpose digital I/O pin
			SC1_DAT	I/O	SmartCard1 DATA pin (SC1_UART_RXD)
28	19	13	PD.9	I/O	General purpose digital I/O pin
			SC1_RST	O	SmartCard1 RST pin
			PWM0_CH3	I/O	PWM0 Channel3 output
29	20	14	PD.10	I/O	General purpose digital I/O pin
			PWM0_CH2	I/O	PWM0 Channel2 output
			TC1	I	Timer1 capture input
30		15	PD.11	I/O	General purpose digital I/O pin
			PWM0_CH1	I/O	PWM0 Channel1 output
			TC0	I	Timer0 capture input
31		16	PD.12	I/O	General purpose digital I/O pin
			CLK_Hz	O	1, 1/2, 1/4, 1/8, 1/16 Hz clock output
			PWM0_CH0	I/O	PWM0 Channel0 output
			TM1	I/O	Timer1 external counter input or Timer1 toggle out
			FCLK0	O	Frequency Divider0 output pin
32	21				NC
33	22		PD.13	I/O	General purpose digital I/O pin
			INT1	I	External interrupt 1 input pin
34	23		PD.14	I/O	General purpose digital I/O pin
35	24		PD.15	I/O	General purpose digital I/O pin

Pin No.			Pin Name	Pin Type	Description
64-pin	48-pin	32-pin			
36	25	17	nRESET	I	External reset input: low active. Setting this pin low will reset chip to initial state. With internal pull-up. **Note:** It is recommended to use 10 kΩ pull-up resistor and 10 μF capacitor on nRESET pin.
37	26	18	LDO_CAP	P	LDO capacitor pin
38	27	19	V_{DD}	P	Power supply for I/O ports and LDO source
39	28	20	PF.0	I/O	General purpose digital I/O pin
			X32I	I	External 32.768 kHz crystal input pin (default)
			TM3	I/O	Timer3 external counter input or Timer3 toggle out.
40	29	21	PF.1	I/O	General purpose digital I/O pin
			X32O	O	External 32.768 kHz crystal output pin (default)
			TM2	I/O	Timer2 external counter input or Timer2 toggle out.
41	30	22	V_{SS}	G	Ground for digital circuit
42	31	23	PF.2	I/O	General purpose digital I/O pin
			XT1_IN	AI	External 4~24 MHz crystal input pin (default)
			UART1_RXD	I	UART1 Data receiver input pin
			TC3	I	Timer3 capture input
			INT1	I	External interrupt1 input pin
43	32	24	PF.3	I/O	General purpose digital I/O pin
			XT1_OUT	AO	External 4~24 MHz crystal output pin
			UART1_TXD	O	UART1 Data transmitter output pin (This pin could be modulated with PWM0 output.)
			TC2	I	Timer2 capture input
			INT0	I	External interrupt0 input pin
44	33		AV_{SS}	G	Ground for ADC and comparators
45	34		PA.0	I/O	General purpose digital I/O pin
			AD0	AI	ADC analog input0
46	35		PA.1	I/O	General purpose digital I/O pin
			AD1	AI	ADC analog input1
			ACMP0_P3	AI	Comparator0 P-end input3
			ACMP0_CHDIS	O	Comparator0 charge/discharge path
47	36		PA.2	I/O	General purpose digital I/O pin

Pin No.			Pin Name	Pin Type	Description
64-pin	48-pin	32-pin			
			SC0_CLK	**O**	SmartCard0 clock pin (SC0_UART_TXD)
			INT0	**I**	External interrupt0 input pin
			AD2	**AI**	ADC analog input2
			ACMP0_P2	**AI**	Comparator0 P-end input2
			ACMP0_CHDIS	**O**	Comparator0 charge/discharge path
48	37		PA.3	**I/O**	General purpose digital I/O pin
			SC0_DAT	**I/O**	SmartCard0 DATA pin(SC0_UART_RXD)
			INT1	**I**	External interrupt 1
			AD3	**AI**	ADC analog input3
			ACMP0_P1	**AI**	Comparator0 P-end input1
			ACMP0_CHDIS	**O**	Comparator0 charge/discharge path
49	38	25	PA.4	**I/O**	General purpose digital I/O pin
			SC0_CD	**I**	SmartCard0 card detect pin
			AD4	**AI**	ADC analog input4
			ACMP0_P0	**AI**	Comparator0 P-end input0
			ACMP0_CHDIS	**O**	Comparator0 charge/discharge path
50	39	26	PA.5	**I/O**	General purpose digital I/O pin
			SPI1_SS0	**I/O**	SPI1 1st slave select pin
			I2C1_SDA	**I/O**	I^{2}C1 data I/O pin
			SC0_PWR	**O**	SmartCard0 Power pin
			AD5	**AI**	ADC analog input5
			ACMP0_N	**AI**	Comparator0 N-end input0
			ACMP0_CHDIS	**O**	Comparator0 charge/discharge path
51	40		PA.6	**I/O**	General purpose digital I/O pin
			ACMP0_CHDIS	**O**	Comparator0 charge/discharge path
			SC0_RST	**O**	SmartCard0 RST pin
			ACMP0_OUT	**O**	Comparator0 output
			AD6	**AI**	ADC analog input6
52	41		V$_{REF}$	**A**	ADC/Comparator reference voltage
53	42	27	AV$_{DD}$	**P**	Power supply for ADC and comparators
54	43	28	PF.4	**I/O**	General purpose digital I/O pin
			ICE_CLK	**I**	Serial Wired Debugger Clock pin **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE_CLK pin.

Pin No.			Pin Name	Pin Type	Description
64-pin	48-pin	32-pin			
			CLK_Hz	**O**	1, 1/2, 1/4, 1/8, 1/16 Hz clock output
			PWM0_CH2	**O**	PWM0 Channel2 output
			TC1	**I**	Timer1 capture input
			FCLK1	**O**	Frequency Divider1 output pin
55	44	29	PF.5	**I/O**	General purpose digital I/O pin
			ICE_DAT	**I/O**	Serial Wired Debugger Data pin **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE_DAT pin.
			PWM0_CH3	**I/O**	PWM0 Channel3 output
			TC0	**I**	Timer0 capture input
			ACMP0_CHDIS	**O**	Comparator0 charge/discharge path
56	45	30	PA.12	**I/O**	General purpose digital I/O pin
			UART0_TXD	**O**	UART0 Data transmitter output pin (This pin could be modulated with PWM0 output.)
			SPI1_MOSI0	**I/O**	SPI1 1st MOSI (Master Out, Slave In) pin
			I2C0_SCL	**I/O**	I^{2}C 0 clock pin
			ACMP1_P	**AI**	Comparator1 P-end input
57	46	31	PA.13	**I/O**	General purpose digital I/O pin
			UART0_RXD	**I**	UART0 Data receiver input pin
			SPI1_MISO0	**I/O**	SPI1 1st MISO (Master In, Slave Out) pin
			I2C0_SDA	**I/O**	I^{2}C0 data I/O pin
			ACMP1_N	**AI**	Comparator1 N-end input
58	47	32	PA.14	**I/O**	General purpose digital I/O pin
			SPI1_CLK	**I/O**	SPI1 serial clock pin
			I2C1_SCL	**I/O**	I^{2}C1 clock pin
			ACMP0_CHDIS	**O**	Comparator0 charge/discharge path
59	48		PA.15	**I/O**	General purpose digital I/O pin
			SPI1_SS0	**I/O**	SPI1 1st slave select pin
			I2C1_SDA	**I/O**	I^{2}C1 data I/O pin
			TC3	**I**	Timer3 capture input
			ACMP1_OUT	**O**	Comparator1 output
60			PB.0	**I/O**	General purpose digital I/O pin
			UART0_TXD	**O**	UART0 Data transmitter output pin(This pin could modulate with PWM0 output)
			FCLK1	**O**	Frequency Divider1 output pin

Pin No.			Pin Name	Pin Type	Description
64-pin	48-pin	32-pin			
61			PB.1	**I/O**	General purpose digital I/O pin
			UART0_RXD	**I**	UART0 Data receiver input pin
			TC2	**I**	Timer 2 capture input
			INT1	**I**	External interrupt1 input pin
62			PB.2	**I/O**	General purpose digital I/O pin
			UART0_RTSn	**O**	UART0 Request to Send output pin
			SPI1_MOSI1	**I/O**	SPI1 2nd MOSI (Master Out, Slave In) pin
			I2C0_SCL	**I/O**	I^{2}C0 clock pin
			TM3	**I/O**	Timer3 external counter input or Timer3 toggle out.
63			PB.3	**I/O**	General purpose digital I/O pin
			UART0_CTSn	**I**	UART0 Clear to Send input pin
			SPI1_MISO1	**I/O**	SPI1 2nd MISO (Master In, Slave Out) pin
			I2C0_SDA	**I/O**	I^{2}C0 data I/O pin
			TM2	**I/O**	Timer2 external counter input or Timer2 toggle out.
64			PB.6	**I/O**	General purpose digital I/O pin
			UART1_TXD	**O**	UART1 Data transmitter output pin (This pin could be modulated with PWM0 output.)
			SPI1_SS1	**I/O**	SPI1 2nd slave select pin
			FCLK0	**O**	Frequency Divider0 output pin

Note: Pin Type: I = Digital Input, O = Digital Output; AI = Analog Input; AO = Analog Output; P = Power Pin; AP = Analog Power.

4.4.2 NuMicro® Nano112 Pin Description

Pin No.			Pin Name	Pin Type	Description
100-pin	64-pin	48-pin			
1			PB.7	I/O	General purpose digital I/O pin
			LCD_SEG33	O	LCD segment output 33 at 100-pin
			UART1_CTSn	I	UART1 Clear to Send input pin
			SC0_CD	I	SmartCard0 card detect
2			PB.8	I/O	General purpose digital I/O pin
			LCD_SEG32	O	LCD segment output 32 at 100-pin
			SNOOPER	I	Snooper pin
			PWM0_CH0	I/O	PWM0 Channel0 output
			TM0	I/O	Timer0 external counter input or Timer0 toggle out.
			INT1	I	External interrupt1 input pin
3			PB.9	I/O	General purpose digital I/O pin
			LCD_SEG31	O	LCD segment output 31 at 100-pin
			PWM0_CH1	I/O	PWM0 Channel1 output
4			PE.8	I/O	General purpose digital I/O pin
			LCD_SEG30	O	LCD segment output 30 at 100-pin
			PWM0_CH2	I/O	PWM0 Channel2 output
5			PE.9	I/O	General purpose digital I/O pin
			LCD_SEG29	O	LCD segment output 29 at 100-pin
			PWM0_CH3	I/O	PWM0 Channel3 output
6	1		PB.10	I/O	General purpose digital I/O pin
			LCD_SEG28	O	LCD segment output 28 at 100-pin
			LCD_SEG24	O	LCD segment output 24 at 64-pin
			UART1_RXD	I	UART1 Data receiver input pin
			SPI0_MOSI1	I/O	SPI0 2nd MOSI (Master Out, Slave In) pin
7	2		PB.11	I/O	General purpose digital I/O pin
			LCD_SEG27	O	LCD segment output 27 at 100-pin
			LCD_SEG23	O	LCD segment output 23 at 64-pin
			UART1_RTSn	O	UART1 Request to Send output pin
			SPI0_MISO1	I/O	SPI0 2rd MISO (Master In, Slave Out) pin
			TM1	I/O	Timer1 external counter input or Timer1 toggle out
8	3	1	PB.12	I/O	General purpose digital I/O pin

Pin No.			Pin Name	Pin Type	Description
100-pin	64-pin	48-pin			
			LCD_SEG26	O	LCD segment output 26 at 100-pin
			LCD_SEG22	O	LCD segment output 22 at 64-pin
			LCD_SEG15	O	LCD segment output 15 at 48-pin
			UART0_RTSn	O	UART0 Request to Send output pin
			SPI0_MOSI0	I/O	SPI0 1st MOSI (Master Out, Slave In) pin
			TM0	I/O	Timer0 external counter input or Timer0 toggle out.
			FCLK0	O	Frequency Divider0 output pin
9	4	2	PB.13	I/O	General purpose digital I/O pin
			LCD_SEG25	O	LCD segment output 25 at 100-pin
			LCD_SEG21	O	LCD segment output 21 at 64-pin
			LCD_SEG14	O	LCD segment output 14 at 48-pin
			UART0_RXD	I	UART0 Data receiver input pin
			SPI0_MISO0	I/O	SPI0 1st MISO (Master In, Slave Out) pin
10	5	3	PB.14	I/O	General purpose digital I/O pin
			LCD_SEG24	O	LCD segment output 24 at 100-pin
			LCD_SEG20	O	LCD segment output 20 at 64-pin
			LCD_SEG13	O	LCD segment output 13 at 48-pin
			UART0_TXD	O	UART0 Data transmitter output pin (This pin could be modulated with PWM0 output.)
			SPI0_CLK	I/O	SPI0 serial clock pin
11			NC		
12	6	4	PB.15	I/O	General purpose digital I/O pin
			LCD_SEG23	O	LCD segment output 23 at 100-pin
			LCD_SEG19	O	LCD segment output 19 at 64-pin
			LCD_SEG12	O	LCD segment output 12 at 48-pin
			UART0_CTSn	I	UART0 Clear to Send input pin
			SPI0_SS0	I/O	SPI0 1st slave select pin
13	7	5	PC.0	I/O	General purpose digital I/O pin
			LCD_SEG22	O	LCD segment output 24 at 100-pin
			LCD_SEG18	O	LCD segment output 18 at 64-pin
			LCD_SEG11	O	LCD segment output 11 at 48-pin
			SPI0_SS1	I/O	SPI0 2nd slave select pin
			I2C0_SCL	I/O	I^{2}C0 clock pin

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

Pin No.			Pin Name	Pin Type	Description
100-pin	64-pin	48-pin			
			PWM0_CH0	I/O	PWM0 Channel0 output
14	8	6	PC.1	I/O	General purpose digital I/O pin
			LCD_SEG21	O	LCD segment output 21 at 100-pin
			LCD_SEG17	O	LCD segment output 17 at 64-pin
			LCD_SEG10	O	LCD segment output 10 at 48-pin
			I2C0_SDA	I/O	I^{2}C0 data I/O pin
			PWM0_CH1	I/O	PWM0 Channel1 output
15	9	7	PC.2	I/O	General purpose digital I/O pin
			LCD_SEG20	O	LCD segment output 20 at 100-pin
			LCD_SEG16	O	LCD segment output 16 at 64-pin
			LCD_SEG9	O	LCD segment output 9 at 48-pin
			I2C1_SCL	O	I^{2}C1 clock pin
			PWM0_CH2	I/O	PWM0 Channel2 output
16	10	8	PC.3	I/O	General purpose digital I/O pin
			LCD_SEG19	O	LCD segment output 19 at 100-pin
			LCD_SEG15	O	LCD segment output 15 at 64-pin
			LCD_SEG8	O	LCD segment output 8 at 48-pin
			I2C1_SDA	I/O	I^{2}C1 data I/O pin
			PWM0_CH3	I/O	PWM0 Channel3 output
17	11	9	PC.4	I/O	General purpose digital I/O pin
			LCD_SEG18	O	LCD segment output 18 at 100-pin
			LCD_SEG14	O	LCD segment output 14 at 64-pin
			LCD_SEG7	O	LCD segment output 7 at 48-pin
			UART1_CTSn	I	UART1 Clear to Send input pin
			SC0_CLK	O	SmartCard0 clock pin (SC0_UART_TXD)
			INT0	I	External interrupt0 input pin
18	12	10	PC.5	I/O	General purpose digital I/O pin
			LCD_SEG17	O	LCD segment output 17 at 100-pin
			LCD_SEG13	O	LCD segment output 13 at 64-pin
			LCD_SEG6	O	LCD segment output 6 at 48-pin
			SC0_CD	I	SmartCard0 card detect pin
19	13	11	PC.6	I/O	General purpose digital I/O pin
			LCD_SEG16	O	LCD segment output 16 at 100-pin

Pin No.			Pin Name	Pin Type	Description
100-pin	64-pin	48-pin			
			LCD_SEG12	O	LCD segment output 12 at 64-pin
			LCD_SEG5	O	LCD segment output 5 at 48-pin
			UART1_RTSn	O	UART1 Request to Send output pin
			SC0_DAT	I/O	SmartCard0 DATA pin (SC0_UART_RXD)
20	14	12	PC.7	I/O	General purpose digital I/O pin
			LCD_SEG15	O	LCD segment output 15 at 100-pin
			LCD_SEG11	O	LCD segment output 11 at 64-pin
			LCD_SEG4	O	LCD segment output 4 at 48-pin
			UART1_RXD	I	UART1 Data receiver input pin
			SC0_PWR	O	SmartCard0 Power pin
21	15	13	PC.8	I/O	General purpose digital I/O pin
			LCD_SEG14	O	LCD segment output 14 at 100-pin
			LCD_SEG10	O	LCD segment output 10 at 64-pin
			LCD_SEG3	O	LCD segment output 3 at 48-pin
			UART1_TXD	O	UART1 Data transmitter output pin (This pin could be modulated with PWM0 output.)
			SC0_RST	O	SmartCard0 RST pin
22	16	14	PC.9	I/O	General purpose digital I/O pin
			LCD_SEG13	O	LCD segment output 13 at 100-pin
			LCD_SEG9	O	LCD segment output 9 at 64-pin
			LCD_SEG2	O	LCD segment output 2 at 48-pin
23			V_{DD}	P	Power supply for I/O ports and LDO source
24			V_{SS}	G	Ground for digital circuit
25			V_{SS}	G	Ground for digital circuit
26			PC.10	I/O	General purpose digital I/O pin
			LCD_SEG12	O	LCD segment output 12 at 100-pin
			I2C1_SCL	I/O	I^{2}C1 clock pin
			SC1_CD	I	SmartCard1 card detect pin
27			PC.11	I/O	General purpose digital I/O pin
			LCD_SEG11	O	LCD segment output 11 at 100-pin
			I2C1_SDA	I/O	I^{2}C 1 data I/O pin
			SC1_PWR	O	SmartCard1 PWR pin
28			PC.12	I/O	General purpose digital I/O pin
			LCD_SEG10	O	LCD segment output 10 at 100-pin

Pin No.			Pin Name	Pin Type	Description
100-pin	64-pin	48-pin			
			SC1_CLK	O	SmartCard1 clock pin (SC1_UART_TXD)
29			PC.13	I/O	General purpose digital I/O pin
			LCD_SEG9	O	LCD segment output 9 at 100-pin
			SC1_DAT	I/O	SmartCard1 DATA pin (SC1_UART_RXD)
30	17	15	PC.14	I/O	General purpose digital I/O pin
			LCD_SEG8	O	LCD segment output 8 at 100-pin
			LCD_SEG8	O	LCD segment output 8 at 64-pin
			LCD_SEG1	O	LCD segment output 1 at 48-pin
			SC1_CD	I	SmartCard1 card detect
31	18	16	PC.15	I/O	General purpose digital I/O pin
			LCD_SEG7	O	LCD segment output 7 at 100-pin
			LCD_SEG7	O	LCD segment output 7 at 64-pin
			LCD_SEG0	O	LCD segment output 0 at 48-pin
			SC1_PWR	O	SmartCard1 PWR pin
32	19		PD.0	I/O	General purpose digital I/O pin
			LCD_SEG6	O	LCD segment output 6 at 100-pin
			LCD_SEG6	O	LCD segment output 6 at 64-pin
33	20		PD.1	I/O	General purpose digital I/O pin
			LCD_SEG5	O	LCD segment output 5 at 100-pin
			LCD_SEG5	O	LCD segment output 5 at 64-pin
34	21		PD.2	I/O	General purpose digital I/O pin
			LCD_SEG4	O	LCD segment output 4 at 100-pin
			LCD_SEG4	O	LCD segment output 4 at 64-pin
35	22		PD.3	I/O	General purpose digital I/O pin
			LCD_SEG3	O	LCD segment output 3 at 100-pin
			LCD_SEG3	O	LCD segment output 3 at 64-pin
36	23		PD.4	I/O	General purpose digital I/O pin
			LCD_SEG2	O	LCD segment output 2 at 100-pin
			LCD_SEG2	O	LCD segment output 2 at 64-pin
			SC1_RST	O	SmartCard1 RST pin
37	24		PD.5	I/O	General purpose digital I/O pin
			LCD_SEG1	O	LCD segment output 1 at 100-pin (or as LCD_COM5)

Pin No.			Pin Name	Pin Type	Description
100-pin	64-pin	48-pin			
			LCD_SEG1	O	LCD segment output 1 at 64-pin (or as LCD_COM5)
38	25		PD.6	I/O	General purpose digital I/O pin
			LCD_SEG0	O	LCD segment output 0 at 100-pin(or as LCD_COM4)
			LCD_SEG0	O	LCD segment output 0 at 64-pin (or as LCD_COM4)
39	26	17	PD.7	I/O	General purpose digital I/O pin
			LCD_COM3	O	LCD common output 3 at 100-pin
			LCD_COM3	O	LCD common output 3 at 64-pin
			LCD_COM3	O	LCD common output 3 at 48-pin
			SC1_CLK	O	SmartCard1 clock pin (SC1_UART_TXD)
40	27	18	PD.8	I/O	General purpose digital I/O pin
			LCD_COM2	O	LCD common output 2 at 100-pin
			LCD_COM2	O	LCD common output 2 at 64-pin
			LCD_COM2	O	LCD common output 2 at 48-pin
			SC1_DAT	I/O	SmartCard1 DATA pin (SC1_UART_RXD)
41	28	19	PD.9	I/O	General purpose digital I/O pin
			LCD_COM1	O	LCD common output 1 at 100-pin
			LCD_COM1	O	LCD common output 1 at 64-pin
			LCD_COM1	O	LCD common output 1 at 48-pin
			SC1_RST	O	SmartCard1 RST pin
			PWM0_CH3	I/O	PWM0 Channel3 output
42	29	20	PD.10	I/O	General purpose digital I/O pin
			LCD_COM0	O	LCD common output 0 at 100-pin
			LCD_COM0	O	LCD common output 0 at 64-pin
			LCD_COM0	O	LCD common output 0 at 48-pin
			PWM0_CH2	I/O	PWM0 Channel2 output
			TC1	I	Timer1 capture input
43	30		PD.11	I/O	General purpose digital I/O pin
			LCD_DH2	O	LCD external capacitor pin of charge pump circuit at 100-pin
			LCD_DH2	O	LCD external capacitor pin of charge pump circuit at 64-pin
			PWM0_CH1	I/O	PWM0 Channel1 output
			TC0	I	Timer0 capture input

Pin No.			Pin Name	Pin Type	Description
100-pin	64-pin	48-pin			
44	31		PD.12	I/O	General purpose digital I/O pin
			CLK_Hz	O	1, 1/2, 1/4, 1/8, 1/16 Hz clock output
			LCD_DH1	O	LCD external capacitor pin of charge pump circuit at 100-pin
			LCD_DH1	O	LCD external capacitor pin of charge pump circuit at 64-pin
			PWM0_CH0	I/O	PWM0 Channel0 output
			TM1	I/O	Timer1 external counter input
			FCLK0	O	Frequency Divider0 output pin
45					NC
46	32	21	V_{LCD}	P	LCD power supply pin
47					NC
48	33	22	PD.13	I/O	General purpose digital I/O pin
			LCD_V1	I	Input pin of the 1st most positive LCD level at 100-pin
			LCD_V1	I	Input pin of the 1st most positive LCD level at 64-pin
			LCD_V1	I	Input pin of the 1st most positive LCD level at 48-pin
			INT1	I	External interrupt 1 input pin
49	34	23	PD.14	I/O	General purpose digital I/O pin
			LCD_V2	I	Input pin of the 2nd most positive LCD level at 100-pin
			LCD_V2	I	Input pin of the 2nd most positive LCD level at 64-pin
			LCD_V2	I	Input pin of the 2nd most positive LCD level at 48-pin
50	35	24	PD.15	I/O	General purpose digital I/O pin
			LCD_V3	I	Input pin of the 3rd most positive LCD level at 100-pin
			LCD_V3	I	Input pin of the 3rd most positive LCD level at 64-pin
			LCD_V3	I	Input pin of the 3rd most positive LCD level at 48-pin
51	35	25	nRESET		External reset input: low active. Setting this pin low will reset chip to initial state. With internal pull-up. **Note:** It is recommended to use 10 kΩ pull-up resistor and 10 µF capacitor on nRESET pin.
52	37	26	LDO_CAP	P	LDO capacitor pin
53	38	27	V_{DD}	P	Power supply for I/O ports and LDO source

Pin No.			Pin Name	Pin Type	Description
100-pin	64-pin	48-pin			
54	38	28	PF.0	I/O	General purpose digital I/O pin
			X32I	I	External 32.768 kHz crystal input pin(default)
			TM3	I/O	Timer3 external counter input or Timer3 toggle out.
55	40	29	PF.1	I/O	General purpose digital I/O pin
			X32O	O	External 32.768 kHz crystal output pin(default)
			TM2	I/O	Timer2 external counter input or Timer2 toggle out.
56			V_{SS_PLL}	G	Ground for PLL
57	41	30	V_{SS}	G	Ground for digital circuit
58			V_{SS}	G	Ground for digital circuit
59	42	31	PF.2	I/O	General purpose digital I/O pin
			XT1_IN	AI	External 4~24 MHz crystal input pin(default)
			UART1_RXD	I	UART1 Data receiver input pin
			TC3	I	Timer3 capture input
			INT1	I	External interrupt1 input pin
60	43	32	PF.3	I/O	General purpose digital I/O pin
			XT1_OUT	AO	External 4~24 MHz crystal output pin
			UART1_TXD	O	UART1 Data transmitter output pin (This pin could be modulated with PWM0 output.)
			TC2	I	Timer 2 capture input
			INT0	I	External interrupt0 input pin
61					NC
62			PE.0	I/O	General purpose digital I/O pin
			SPI0_MOSI0	I/O	SPI0 1st MOSI (Master Out, Slave In) pin
63			PE.1	I/O	General purpose digital I/O pin
			SPI0_MISO0	I/O	SPI0 1st MISO (Master In, Slave Out) pin
64			PE.2	I/O	General purpose digital I/O pin
			SPI0_CLK	I/O	SPI0 serial clock pin
65			PE.3	I/O	General purpose digital I/O pin
			SPI0_SS0	I/O	SPI0 1st slave select pin
66			PE.4	I/O	General purpose digital I/O pin
			SC1_RST	O	SmartCard1 RST pin
67			PE.5	I/O	General purpose digital I/O pin
			SC1_PWR	O	SmartCard1 PWR pin

Pin No.			Pin Name	Pin Type	Description
100-pin	64-pin	48-pin			
68			PE.6	I/O	General purpose digital I/O pin
			SC1_CLK	O	SmartCard1 clock pin (SC1_UART_TXD)
69			PE.7	I/O	General purpose digital I/O pin
			SC1_DAT	I/O	SmartCard1 DATA pin (SC1_UART_RXD)
70	44	33	AV$_{SS}$	G	Ground for ADC and comparators
71			AV$_{SS}$	G	Ground for ADC and comparators
72	45	34	PA.0	I/O	General purpose digital I/O pin
			AD0	AI	ADC analog input0
73	46	35	PA.1	I/O	General purpose digital I/O pin
			ACMP0_CHDIS	O	Comparator0 charge/discharge path
			ACMP0_P3	AI	Comparator0 P-end input3
			AD1	AI	ADC analog input1
74	47	36	PA.2	I/O	General purpose digital I/O pin
			ACMP0_CHDIS	O	Comparator0 charge/discharge path
			SC0_CLK	O	SmartCard0 clock pin (SC0_UART_TXD)
			ACMP0_P2	AI	Comparator0 P-end input2
			AD2	AI	ADC analog input2
			INT0	I	External interrupt0 input pin
75	48	37	PA.3	I/O	General purpose digital I/O pin
			ACMP0_CHDIS	O	Comparator0 charge/discharge path
			SC0_DAT	I/O	SmartCard0 DATA pin (SC0_UART_RXD)
			ACMP0_P1	AI	Comparator0 P-end input1
			AD3	AI	ADC analog input3
			INT1	I	External interrupt 1
76	49	38	PA.4	I/O	General purpose digital I/O pin
			ACMP0_CHDIS	O	Comparator0 charge/discharge path
			SC0_CD	I	SmartCard0 card detect pin
			ACMP0_P0	AI	Comparator0 P-end input0
			AD4	AI	ADC analog input4
77	50	39	PA.5	I/O	General purpose digital I/O pin
			ACMP0_CHDIS	O	Comparator0 charge/discharge path
			SPI1_SS0	I/O	SPI1 1st slave select pin
			I2C1_SDA	I/O	I^{2}C1 data I/O pin

Pin No.			Pin Name	Pin Type	Description
100-pin	64-pin	48-pin			
			SC0_PWR	O	SmartCard0 Power pin
			ACMP0_N	AI	Comparator0 N-end input0
			AD5	AI	ADC analog input5
78	51	40	PA.6	I/O	General purpose digital I/O pin
			ACMP0_CHDIS	O	Comparator0 charge/discharge path
			SC0_RST	O	SmartCard0 RST pin
			ACMP0_OUT	O	Comparator0 output
			AD6	AI	ADC analog input6
79			PA.7	I/O	General purpose digital I/O pin
			SC1_CD	I	SmartCard1 card detect
			AD7	AI	ADC analog input7
80	52	41	V_{REF}	A	ADC/Comparator reference voltage
81	53	42	AV_{DD}	P	Power supply for ADC and comparators
82	54	43	PF.4	I/O	General purpose digital I/O pin
			ICE_CLK	I	Serial Wired Debugger Clock pin **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE_CLK pin.
			CLK_Hz	O	1, 1/2, 1/4, 1/8, 1/16 Hz clock output
			PWM0_CH2	O	PWM0 Channel2 output
			TC1	I	Timer1 capture input
			FCLK1	O	Frequency Divider1 output pin
83	55	44	PF.5	I/O	General purpose digital I/O pin
			ICE_DAT	I/O	Serial Wired Debugger Data pin **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE_DAT pin.
			ACMP0_CHDIS	O	Comparator0 charge/discharge path
			PWM0_CH3	I/O	PWM0 Channel3 output
			TC0	I	Timer0 capture input
84			PA.8	I/O	General purpose digital I/O pin
			SC0_PWR	O	SmartCard0 Power pin
85			PA.9	I/O	General purpose digital I/O pin
			SC0_RST	O	SmartCard0 RST pin
86			PA.10	I/O	General purpose digital I/O pin
			SC0_CLK	O	SmartCard0 clock pin (SC0_UART_TXD)
87			PA.11	I/O	General purpose digital I/O pin

Pin No.			Pin Name	Pin Type	Description
100-pin	64-pin	48-pin			
			SC0_DAT	I/O	SmartCard0 DATA pin(SC0_UART_RXD)
			STADC	I	ADC external trigger input.
88	56	45	PA.12	I/O	General purpose digital I/O pin
			LCD_SEG19	O	LCD segment output 19 at 48-pin
			UART0_TXD	O	UART0 Data transmitter output pin (This pin could be modulated with PWM0 output.)
			SPI1_MOSI0	I/O	SPI1 1st MOSI (Master Out, Slave In) pin
			I2C0_SCL	I/O	I^{2}C 0 clock pin
			ACMP1_P	AI	Comparator1 P-end input
89	57	46	PA.13	I/O	General purpose digital I/O pin
			LCD_SEG18	O	LCD segment output 18 at 48-pin
			UART0_RXD	I	UART0 Data receiver input pin
			SPI1_MISO0	I/O	SPI1 1st MISO (Master In, Slave Out) pin
			I2C0_SDA	I/O	I^{2}C0 data I/O pin
			ACMP1_N	AI	Comparator1 N-end input
90	58	47	PA.14	I/O	General purpose digital I/O pin
			ACMP0_CHDIS	O	Comparator0 charge/discharge path
			LCD_SEG31	O	LCD segment output 31 at 64-pin
			LCD_SEG17	O	LCD segment output 17 at 48-pin
			SPI1_CLK	I/O	SPI1 serial clock pin
			I2C1_SCL	I/O	I^{2}C1 clock pin
91	59	48	PA.15	I/O	General purpose digital I/O pin
			LCD_SEG30	O	LCD segment output 30 at 64-pin
			LCD_SEG16	O	LCD segment output 16 at 48-pin
			SPI1_SS0	I/O	SPI1 1st slave select pin
			I2C1_SDA	I/O	I^{2}C1 data I/O pin
			ACMP1_OUT	O	Comparator1 output
			TC3	I	Timer3 capture input
92	60		PB.0	I/O	General purpose digital I/O pin
			LCD_SEG29	O	LCD segment output 29 at 64-pin
			UART0_TXD	O	UART0 Data transmitter output pin (This pin could be modulated with PWM0 output.)
			FCLK1	O	Frequency Divider1 output pin
93	61		PB.1	I/O	General purpose digital I/O pin

Pin No.			Pin Name	Pin Type	Description
100-pin	64-pin	48-pin			
			LCD_SEG28	O	LCD segment output 28 at 64-pin
			UART0_RXD	I	UART0 Data receiver input pin
			TC2	I	Timer 2 capture input
			INT1	I	External interrupt1 input pin
94	62		PB.2	I/O	General purpose digital I/O pin
			LCD_SEG27	O	LCD segment output 27 at 64-pin
			UART0_RTSn	O	UART0 Request to Send output pin
			SPI1_MOSI1	I/O	SPI1 2nd MOSI (Master Out, Slave In) pin
			I2C0_SCL	O	I^{2}C0 clock pin
			TM3	I/O	Timer3 external counter input or Timer3 toggle out.
95	63		PB.3	I/O	General purpose digital I/O pin
			LCD_SEG26	O	LCD segment output 26 at 64-pin
			UART0_CTSn	I	UART0 Clear to Send input pin
			SPI1_MISO1	I/O	SPI1 2nd MISO (Master In, Slave Out) pin
			I2C0_SDA	I/O	I^{2}C0 data I/O pin
			TM2	I/O	Timer2 external counter input or Timer2 toggle out.
96			V_{DD}	P	Power supply for I/O ports and LDO source
97			V_{SS}	G	Ground for digital circuit
98			PB.4	I/O	General purpose digital I/O pin
			UART1_RTSn	O	UART1 Request to Send output pin
			SPI1_MISO1	I/O	SPI1 2nd MISO (Master In, Slave Out) pin
99			PB.5	I/O	General purpose digital I/O pin
			LCD_SEG35	O	LCD segment output 35 at 100-pin
			UART1_RXD	I	UART1 Data receiver input pin
			SPI1_MOSI1	I/O	SPI1 2nd MOSI (Master Out, Slave In) pin
100	64		PB.6	I/O	General purpose digital I/O pin
			LCD_SEG34	O	LCD segment output 34 at 100-pin
			LCD_SEG25	O	LCD segment output 25 at 64-pin
			UART1_TXD	O	UART1 Data transmitter output pin (This pin could be modulated with PWM0 output.)
			SPI1_SS1	I/O	SPI1 2nd slave select pin
			FCLK0	O	Frequency Divider0 output pin

Note: Pin Type: I = Digital Input, O=Digital Output; AI=Analog Input; AO= Analog Output; P=Power Pin; AP=Analog Power.

5 BLOCK DIAGRAM

5.1 Nano102 Block Diagram

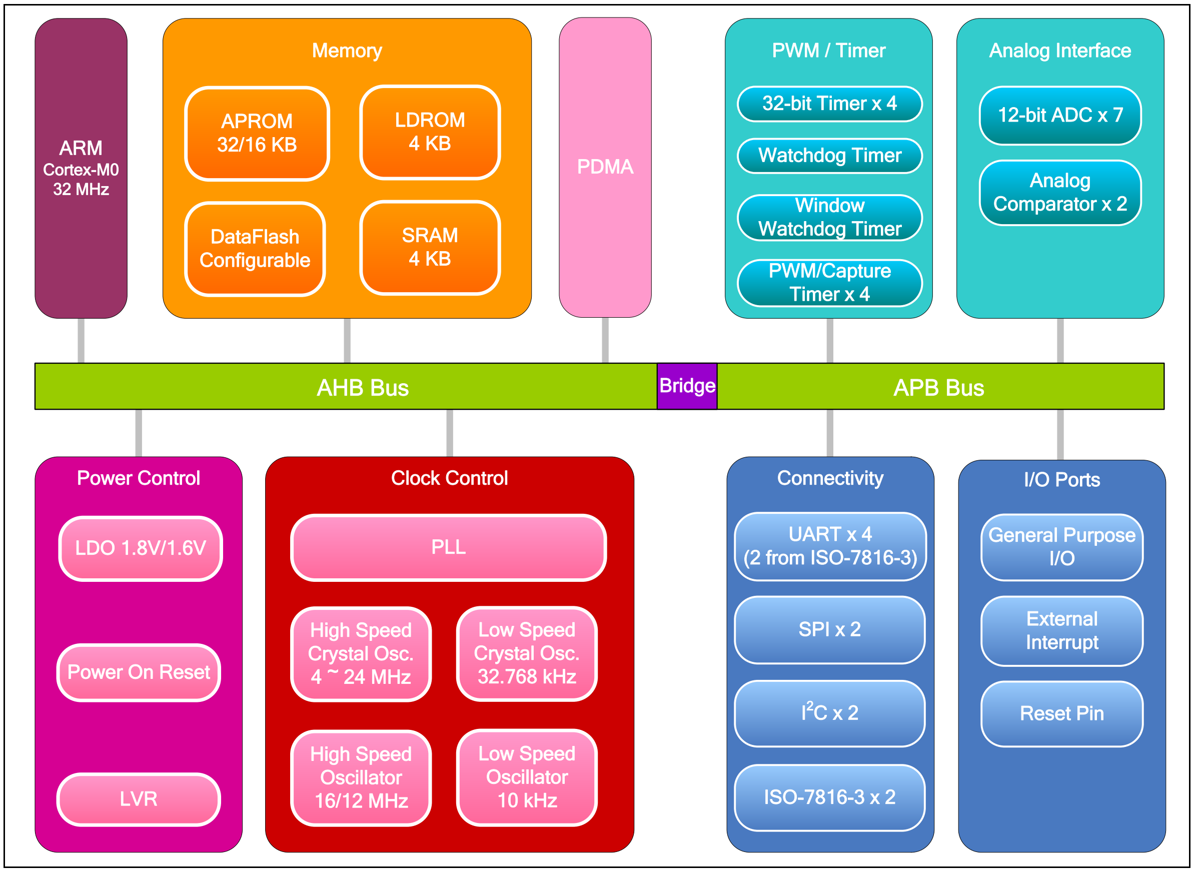


Figure 5-1 NuMicro® Nano102 Block Diagram

5.2 Nano112 Block Diagram

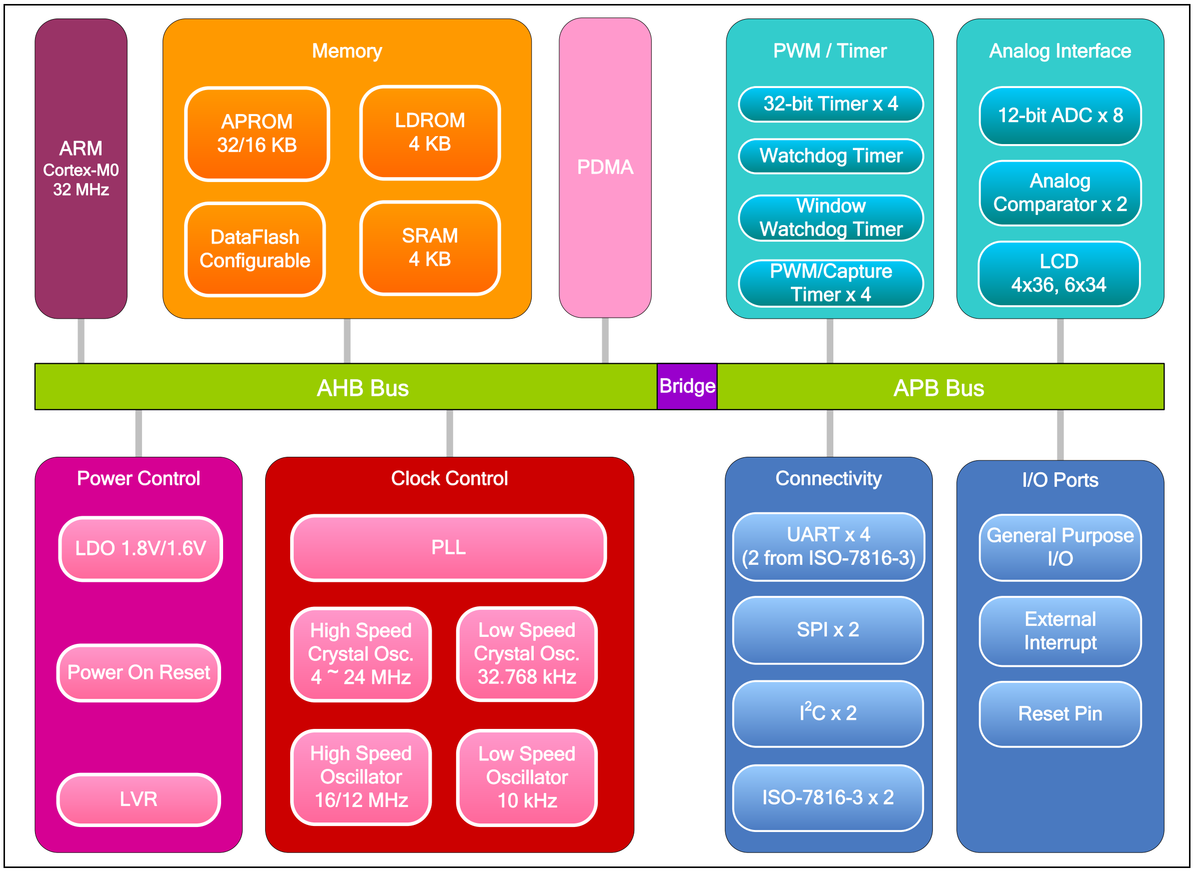


Figure 5-2 NuMicro® Nano112 Block Diagram

6 FUNCTIONAL DESCRIPTION

6.1 ARM® Cortex®-M0 Core

The Cortex®-M0 processor is a configurable, multistage, 32-bit RISC processor, which has an AMBA AHB-Lite interface and includes an NVIC component. It also has optional hardware debug functionality. The processor can execute Thumb code and is compatible with other Cortex®-M profile processor. The profile supports two modes –Thread mode and Handler mode. Handler mode is entered as a result of an exception. An exception return can only be issued in Handler mode. Thread mode is entered on Reset, and can be entered as a result of an exception return. The following figure shows the functional controller of processor.

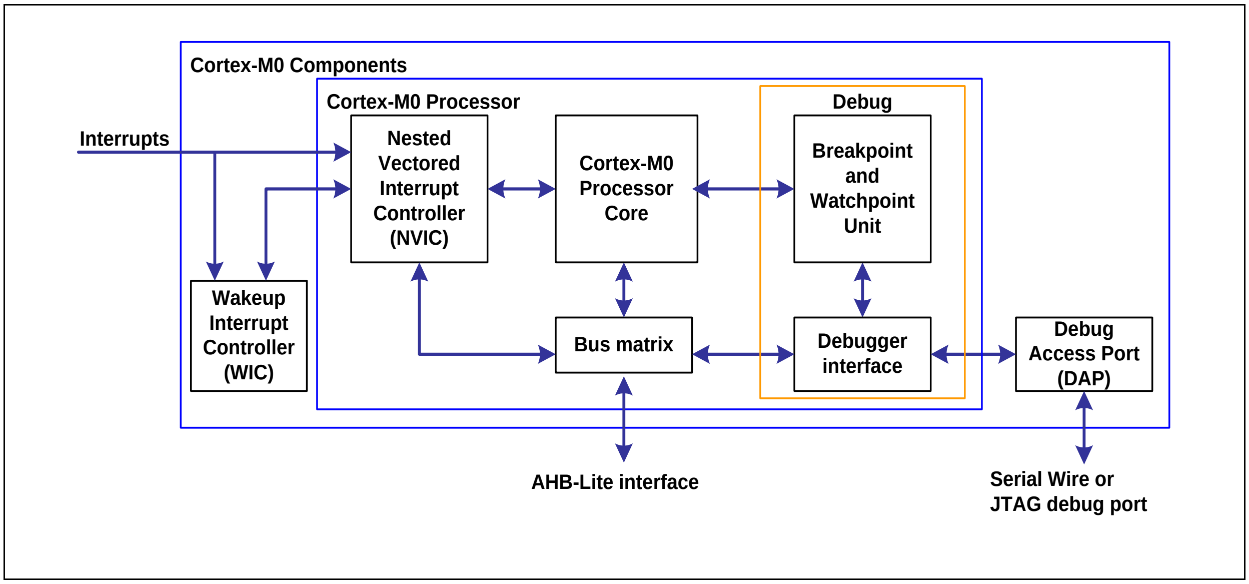


Figure 6-1 Functional Block Diagram

The implemented device provides:

- A low gate count processor:
  - ARMv6-M Thumb® instruction set
  - Thumb-2 technology
  - ARMv6-M compliant 24-bit SysTick timer
  - A 32-bit hardware multiplier
  - System interface supported with little-endian data accesses
  - Ability to have deterministic, fixed-latency, interrupt handling
  - Load/store-multiples and multicycle-multiplies that can be abandoned and restarted to facilitate rapid interrupt handling
  - C Application Binary Interface compliant exception model. This is the ARMv6-M, C Application Binary Interface (C-ABI) compliant exception model that enables the use of pure C functions as interrupt handlers
  - Low Power Sleep mode entry using the Wait For Interrupt (WFI), Wait For Event (WFE)

instructions, or return from interrupt sleep-on-exit feature

- NVIC:
  - 32 external interrupt inputs, each with four levels of priority
  - Dedicated Non-maskable Interrupt (NMI) input
  - Supports for both level-sensitive and pulse-sensitive interrupt lines
  - Supports Wake-up Interrupt Controller (WIC) and, providing Ultra-low Power Sleep mode
- Debug support:
  - Four hardware breakpoints
  - Two watchpoints
  - Program Counter Sampling Register (PCSR) for non-intrusive code profiling
  - Single step and vector catch capabilities
- Bus interfaces:
  - Single 32-bit AMBA-3 AHB-Lite system interface that provides simple integration to all system peripherals and memory
  - Single 32-bit slave port that supports the DAP (Debug Access Port)

6.1.1 System Timer (SysTick)

The Cortex-M0 includes an integrated system timer, SysTick. SysTick provides a simple, 24-bit

cleared-on-write, decrementing, wrap-on-zero counter with a flexible control mechanism. The counter can be used in several different ways, for example:

An RTOS tick timer which fires at a programmable rate (for example 100Hz) and invokes a SysTick routine.

A high speed alarm timer using Core clock.

A variable rate alarm or signal timer – the duration range dependent on the reference clock used and the dynamic range of the counter.

A simple counter. Software can use this to measure task completion time.

An internal Clock Source control based on missing/meeting durations. The COUNTFLAG bit-field in the control and status register can be used to determine if an action completed within a set duration, as part of a dynamic clock management control loop.

When enabled, the timer will count down from the value in the SysTick Current Value Register (SYST_CVR) to zero, and reload (wrap) to the value in the SysTick Reload Value Register (SYST_RVR) on the next clock edge, and then decrement on subsequent clocks. When the counter transitions to zero, the COUNTFLAG status bit is set. The COUNTFLAG bit clears on read.

The SYST_CVR value is UNKNOWN on reset. Software should write to the register to clear it to zero before enabling the feature. This ensures the timer will count from the SYST_RVR value rather than an arbitrary value when it is enabled.

If the SYST_RVR is zero, the timer will be maintained with a current value of zero after it is reloaded with this value. This mechanism can be used to disable the feature independently from the timer enable bit.

For more detailed information, please refer to the “ARM® Cortex®-M0 Technical Reference Manual” and “ARM® v6-M Architecture Reference Manual”.

6.1.2 System Timer Control Register Map

R: read only, W: write only, R/W: both read and write, W&C: Write 1 clear

Register	Offset	R/W	Description	Reset Value
SCS Base Address: **SCS_BA = 0xE000_E000**				
SYST_CTL	SCS_BA+0x10	R/W	SysTick Control and Status	0x0000_0004
SYST_RVR	SCS_BA+0x14	R/W	SysTick Reload Value Register	0xXXXX_XXXX
SYST_CVR	SCS_BA+0x18	R/W	SysTick Current Value Register	0xXXXX_XXXX

6.1.3 System Timer Control Register Description

SysTick Control and Status （SYST_CTL）

Register	Offset	R/W	Description	Reset Value
SYST_CTL	SCS_BA+0x10	R/W	SysTick Control and Status	0x0000_0004

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							COUNTFLAG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					CLKSRC	TICKINT	ENABLE

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**COUNTFLAG**	**System Tick Counter Flag** Returns 1 If Timer Counted to 0 Since Last Time this Register Was Read 0 = COUNTFLAG is cleared on read or by a write to the Current Value register. 1 = COUNTFLAG is set by a count transition from 1 to 0.
[15:3]	**Reserved**	Reserved.
[2]	**CLKSRC**	**System Tick Clock Source Selection** 0 = Clock Source is (optional) external reference clock. 1 = Core clock used for SysTick. If no external clock provided, this bit will read as 1 and ignore writes.
[1]	**TICKINT**	**System Tick Interrupt Enable Control**

		0 = Counting down to 0 does not cause the SysTick exception to be pended. Software can use COUNTFLAG to determine if a count to zero has occurred. 1 = Counting down to 0 will cause the SysTick exception to be pended. Clearing the SysTick Current Value register by a register write in software will not cause SysTick to be pended.
[0]	**ENABLE**	**System Tick Counter Enable Control** 0 = The counter Disabled. 1 = The counter will operate in a multi-shot manner.

SysTick Reload Value Register)SYST_RVR）

Register	Offset	R/W	Description	Reset Value
SYST_RVR	SCS_BA+0x14	R/W	SysTick Reload Value Register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RELOAD							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RELOAD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RELOAD							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**RELOAD**	The value to load into the Current Value register when the counter reaches 0.

SysTick Current Value Register （SYST_CVR）

Register	Offset	R/W	Description	Reset Value
SYST_CVR	SCS_BA+0x18	R/W	SysTick Current Value Register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CURRENT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CURRENT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CURRENT							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**CURRENT**	**System Tick Current Counter Value** This is the value of the counter at the time it is sampled. The counter does not provide read-modify-write protection. The register is write-clear. A software write of any value will clear the register to 0. Unsupported bits RAZ (Read As Zero, writes ignore) (See SysTick Reload Value register).

6.1.4 System Control Registers

Key control and status features of Coterx-M0 are managed centrally in a System Control Block within the System Control Registers.

For more detailed information, please refer to the “ARM® Cortex®-M0 Technical Reference Manual” and “ARM® v6-M Architecture Reference Manual”.

6.1.5 System Control Register Memory Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
SCS Base Address: **SCS_BA = 0xE000_E000**				
CPUID	SCS_BA+0xD00	R	CPUID Base Register	0x410C_C200
ICSR	SCS_BA+0xD04	R/W	Interrupt Control State Register	0x0000_0000
SCR	SCS_BA+0xD10	R/W	System Control Register	0x0000_0000
SHPR2	SCS_BA+0xD1C	R/W	System Handler Priority Register 2	0x0000_0000
SHPR3	SCS_BA+0xD20	R/W	System Handler Priority Register 3	0x0000_0000

6.1.6 System Control Register Description

CPUID Base Register (CPUID)

Register	Offset	R/W	Description	Reset Value
CPUID	SCS_BA+0xD00	R	CPUID Base Register	0x410C_C200

31	30	29	28	27	26	25	24
IMPLEMENTER							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				**PART**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PARTNO							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PARTNO				**REVISION**			

Bits	Description	
[31:24]	**IMPLEMENTER**	**Implementer Code** Implementer code assigned by ARM (ARM = 0x41).
[23:20]	**Reserved**	Reserved.
[19:16]	**PART**	**Architecture of the Processor** Reads as 0xC for ARMv6-M parts
[15:4]	**PARTNO**	**Part Number of the Processor** Reads as 0xC20.
[3:0]	**REVISION**	**Revision Number** Reads as 0x0

Interrupt Control State Register (ICSR)

Register	Offset	R/W	Description	Reset Value
ICSR	SCS_BA+0xD04	R/W	Interrupt Control State Register	0x0000_0000

31	30	29	28	27	26	25	24
NMIPENDSET	**Reserved**		**PENDSVSET**	**PENDSVCLR**	**PENDSTSET**	**PENDSTCLR**	**Reserved**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ISRPREEMPT	**ISRPENDING**	**Reserved**	**VECTPENDING**				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VECTPENDING				**Reserved**			**VECTACTIVE**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VECTACTIVE							

Bits	Description	
[31]	**NMIPENDSET**	**NMI Set-pending Bit** Write Operation: 0 = No effect. 1 = Changes NMI exception state to pending. Read Operation: 0 = NMI exception not pending. 1 = NMI exception pending. **Note:** Because NMI is the highest-priority exception, normally the processor enters the NMI exception handler as soon as it detects a write of 1 to this bit. Entering the handler then clears this bit to 0. This means a read of this bit by the NMI exception handler returns 1 only if the NMI signal is reasserted while the processor is executing that handler.
[30:29]	**Reserved**	Reserved
[28]	**PENDSVSET**	**PendSV Set-pending Bit** Write Operation: 0 = No effect. 1 = Changes PendSV exception state to pending. Read Operation: 0 = PendSV exception is not pending. 1 = PendSV exception is pending. **Note:** Writing 1 to this bit is the only way to set the PendSV exception state to pending
[27]	**PENDSVCLR**	**PendSV Clear-pending Bit** Write Operation: 0 = No effect. 1 = Removes the pending state from the PendSV exception. This bit is write-only. To clear the PENDSV bit, you must “write 0 to PENDSVSET and write 1 to PENDSVCLR” at the same time.
[26]	**PENDSTSET**	**SysTick Exception Set-pending Bit**

		Write Operation: 0 = No effect. 1 = Changes SysTick exception state to pending. Read Operation: 0 = SysTick exception is not pending. 1 = SysTick exception is pending.
[25]	**PENDSTCLR**	**SysTick Exception Clear-pending Bit** Write Operation: 0 = No effect. 1 = Removes the pending state from the SysTick exception. **Note:** This bit is write-only. When you want to clear PENDST bit, you must "write 0 to PENDSTSET and write 1 to PENDSTCLR" at the same time.
[24]	**Reserved**	Reserved
[23]	**ISRPREEMPT**	**Interrupt Preempt Bit (Read Only)** If set, a pending exception will be serviced on exit from the debug halt state.
[22]	**ISRPENDING**	**Interrupt Pending Flag, excluding NMI and Faults (Read Only)** 0 = Interrupt not pending. 1 = Interrupt pending.
[21]	**Reserved**	Reserved
[20:12]	**VECTPENDING**	**Exception Number of the Highest Priority Pending Enabled Exception** 0 = No pending exceptions. Non-zero = Exception number of the highest priority pending enabled exception.
[11:9]	**Reserved**	Reserved
[8:0]	**VECTACTIVE**	**Contains the Active Exception Number** 0 = Thread mode. Non-zero = Exception number of the currently active exception.

System Control Register (SCR)

Register	Offset	R/W	Description	Reset Value
SCR	SCS_BA+0xD10	R/W	System Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			SEVONPEND	Reserved	SLEEPDEEP	SLEEPONEXIT	Reserved

Bits	Description	
[31:5]	Reserved	Reserved
[4]	SEVONPEND	**Send Event on Pending Bit** 0 = Only enabled interrupts or events can wake-up the processor, disabled interrupts are excluded. 1 = Enabled events and all interrupts, including disabled interrupts, can wake-up the processor. When an event or interrupt enters pending state, the event signal wakes up the processor from WFE. If the processor is not waiting for an event, the event is registered and affects the next WFE. The processor also wakes up on execution of an SEV instruction or an external event.
[3]	Reserved	Reserved.
[2]	SLEEPDEEP	**Processor Deep Sleep and Sleep Mode Selection** Controls whether the processor uses sleep or deep sleep as its low power mode: 0 = Sleep mode. 1 = Deep Sleep mode.
[1]	SLEEPONEXIT	**Sleep-on-exit Enable Control** This bit indicates sleep-on-exit when returning from Handler mode to Thread mode. 0 = Do not sleep when returning to Thread mode. 1 = Enter Sleep or Deep Sleep when returning from ISR to Thread mode. Setting this bit to 1 enables an interrupt driven application to avoid returning to an empty main application.

System Handler Priority Register 2 (SHPR2)

Register	Offset	R/W	Description	Reset Value
SHPR2	SCS_BA+0xD1C	R/W	System Handler Priority Register 2	0x0000_0000

31	30	29	28	27	26	25	24
PRI_11							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bits	Description	
[31:30]	**PRI_11**	**Priority of System Handler 11 – SVCall** “0” denotes the highest priority and “3” denotes the lowest priority.

System Handler Priority Register 3 (SHPR3)

Register	Offset	R/W	Description	Reset Value
SHPR3	SCS_BA+0xD20	R/W	System Handler Priority Register 3	0x0000_0000

31	30	29	28	27	26	25	24
PRI_15		**Reserved**					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRI_14		**Reserved**					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:30]	**PRI_15**	**Priority of System Handler 15 – SysTick** “0” denotes the highest priority and “3” denotes the lowest priority.
[23:22]	**PRI_14**	**Priority of System Handler 14 – PendSV** “0” denotes the highest priority and “3” denotes the lowest priority.

6.2 Memory Organization

6.2.1 Overview

The Nano112 provides 4G-byte addressing space. The memory locations assigned to each on-chip modules are shown in following. The detailed register definition, memory space, and programming detailed will be described in the following sections for each on-chip module. The Nano112 series only supports little-endian data format.

6.2.2 Memory Map

The memory locations assigned to each on-chip controllers are shown in the following table.

Address Space	Token	Modules
Flash & SRAM Memory Space		
0x0000_0000 – 0x0001_FFFF	FLASH_BA	FLASH Memory Space (128KB)
0x2000_0000 – 0x2000_3FFF	SRAM_BA	SRAM Memory Space (16KB)
0x6000_0000 --- 0x6001_FFFF	EXTMEM_BA	External Memory Space(128KB)
AHB Modules Space (0x5000_0000 – 0x501F_FFFF)		
0x5000_0000 – 0x5000_01FF	GCR_BA	System Management Control Registers
0x5000_0200 – 0x5000_02FF	CLK_BA	Clock Control Registers
0x5000_0300 – 0x5000_03FF	INT_BA	Interrupt Multiplexer Control Registers
0x5000_4000 – 0x5000_7FFF	GPIO_BA	GPIO Control Registers
0x5000_8000 – 0x5000_BFFF	DMA_BA	DMA Control Registers
0x5000_C000 – 0x5000_FFFF	FMC_BA	Flash Memory Control Registers
APB1 Modules Space (0x4000_0000 ~ 0x400F_FFFF)		
0x4000_4000 – 0x4000_7FFF	WDT_BA	Watchdog Timer Control Registers
0x4000_8000 – 0x4000_BFFF	RTC_BA	Real Time Clock (RTC) Control Register
0x4001_0000 – 0x4001_3FFF	TMR01_BA	Timer0 and Timer1 Control Registers
0x4002_0000 – 0x4002_3FFF	I2C0_BA	I^{2}C0 Interface Control Registers
0x4003_0000 – 0x4003_3FFF	SPI0_BA	SPI0 with Master/Slave function Control Registers
0x4004_0000 – 0x4004_3FFF	PWM0_BA	PWM0 Control Registers
0x4005_0000 – 0x4005_3FFF	UART0_BA	UART0 Control Registers
0x400B_0000 – 0x400B_3FFF	LCD_BA	LCD Control Registers
0x400E_0000 – 0x400E_3FFF	ADC_BA	12-bit Analog-Digital-Converter (ADC) Control Registers
APB2 Modules Space (0x4010_0000 ~ 0x401F_FFFF)		
0x4011_0000 – 0x4011_3FFF	TMR23_BA	Timer2 and Timer3 Control Registers
0x4012_0000 – 0x4012_3FFF	I2C1_BA	I^{2}C1 Interface Control Registers
0x4013_0000 – 0x4013_3FFF	SPI1_BA	SPI1 with Master/Slave function Control Registers

0x4015_0000 – 0x4015_3FFF	UART1_BA	UART1 Control Registers
0x4019_0000 – 0x4019_3FFF	SC0_BA	SmartCard0 Control Registers
0x401B_0000 – 0x401B_3FFF	SC1_BA	SmartCard1 Control Registers
0x401D_0000 – 0x401D_3FFF	ACMP_BA	Analog Comparator Control Registers
System Control Space (0xE000_E000 ~ 0xE000_EFFF)		
0xE000_E010 – 0xE000_E0FF	SCS_BA	System Timer Control Registers
0xE000_E100 – 0xE000_ECFF	SCS_BA	External Interrupt Controller Control Registers
0xE000_ED00 – 0xE000_ED8F	SCS_BA	System Control Registers

6.3 Nested Vectored Interrupt Controller (NVIC)

6.3.1 Overview

The Cortex®-M0 provides an interrupt controller as an integral part of the exception mode, named as "Nested Vectored Interrupt Controller (NVIC)". It is closely coupled to the processor kernel and provides following features:

6.3.2 Features

- Nested and Vectored interrupt support
- Automatic processor state saving and restoration
- Dynamic priority changing
- Reduced and deterministic interrupt latency

The NVIC prioritizes and handles all supported exceptions. All exceptions are handled in "Handler Mode". This NVIC architecture supports 32 (IRQ[31:0]) discrete interrupts with 4 levels of priority. All of the interrupts and most of the system exceptions can be configured to different priority levels. When an interrupt occurs, the NVIC will compare the priority of the new interrupt to the current running one's priority. If the priority of the new interrupt is higher than the current one, the new interrupt handler will override the current handler.

When any interrupts is accepted, the starting address of the interrupt service routine (ISR) is fetched from a vector table in memory. There is no need to determine which interrupt is accepted and branch to the starting address of the correlated ISR by software. While the starting address is fetched, NVIC will also automatically save processor state including the registers "PC, PSR, LR, R0~R3, R12" to the stack. At the end of the ISR, the NVIC will restore the mentioned registers from stack and resume the normal execution. Thus it will take less and deterministic time to process the interrupt request.

The NVIC supports "Tail Chaining" which handles back-to-back interrupts efficiently without the overhead of states saving and restoration and therefore reduces delay time in switching to pending ISR at the end of current ISR. The NVIC also supports "Late Arrival" which improves the efficiency of concurrent ISRs. When a higher priority interrupt request occurs before the current ISR starts to execute (at the stage of state saving and starting address fetching), the NVIC will give priority to the higher one without delay penalty. Thus it advances the real-time capability.

For more detailed information, please refer to the "ARM® Cortex®-M0 Technical Reference Manual" and "ARM® v6-M Architecture Reference Manual".

6.3.3 Exception Model and System Interrupt Map

The following table lists the exception model supported by Nano112 serials. Software can set four levels of priority on some of these exceptions as well as on all interrupts. The highest user-configurable priority is denoted as "0" and the lowest priority is denoted as "3". The default priority of all the user-configurable interrupts is "0". Note that priority "0" is treated as the fourth priority on the system, after three system exceptions "Reset", "NMI" and "Hard Fault".

Exception Name	Vector Number	Priority
Reset	1	-3
NMI	2	-2
Hard Fault	3	-1
Reserved	4 ~ 10	Reserved
SVCall	11	Configurable
Reserved	12 ~ 13	Reserved
PendSV	14	Configurable
SysTick	15	Configurable
Interrupt (IRQ0 ~ IRQ31)	16 ~ 47	Configurable

Table 6‑1 Exception Model

Vector Number	Interrupt Number (Bit In Interrupt Registers)	Interrupt Name	Source IP	Interrupt Description
0 ~ 15	-	-	-	System exceptions
16	0	BOD_INT	Brown-out	Brown-out low voltage detected interrupt
17	1	WDT_INT	WDT	Watchdog Timer interrupt
18	2	EINT0	GPIO	External signal interrupt from GPA.2, GPC.4 or PF.3 pin
19	3	EINT1	GPIO	External signal interrupt from PA.3, PB.1, PB.8, PD.13 or PF.2 pin
20	4	GPABC_INT	GPIO	External signal interrupt from PA[15:0] / PB[13:0]/PC[15:0]
21	5	GPDEF_INT	GPIO	External interrupt from PD[15:0]/PE[9:0]/PF[5:0]
22	6	PWM0_INT	PWM0	PWM0 interrupt
23	7			Reserved
24	8	TMR0_INT	TMR0	Timer0 interrupt
25	9	TMR1_INT	TMR1	Timer1 interrupt
26	10	TMR2_INT	TMR2	Timer2 interrupt
27	11	TMR3_INT	TMR3	Timer3 interrupt
28	12	UART0_INT	UART0	UART0 interrupt
29	13	UART1_INT	UART1	UART1 interrupt
30	14	SPI0_INT	SPI0	SPI0 interrupt
31	15	SPI1_INT	SPI1	SPI1 interrupt
32	16			Reserved
33	17	IRC_INT	IRC	IRC TRIM interrupt

Vector Number	Interrupt Number (Bit In Interrupt Registers)	Interrupt Name	Source IP	Interrupt Description
34	18	I2C0_INT	I2C0	I^{2}C0 interrupt
35	19	I2C1_INT	I2C1	I^{2}C1 interrupt
36	20			Reserved
37	21	SC0_INT	SC0	Smart Card0 interrupt
38	22	SC1_INT	SC1	Smart Card1 interrupt
39	23			Reserved
40	24			Reserved
41	25	LCD_INT	LCD	LCD interrupt
42	26	DMA_INT	DMA	DMA interrupt
43	27			Reserved
44	28	PD_WU_INT	CLKC	Clock controller interrupt for chip wake-up from power-down state
45	29	ADC_INT	ADC	ADC interrupt
46	30			Reserved
47	31	RTC_INT	RTC	Real time clock interrupt

Table 6-2 System Interrupt Map

6.3.4 Vector Table

When any interrupts is accepted, the processor will automatically fetch the starting address of the interrupt service routine (ISR) from a vector table in memory. For ARMv6-M, the vector table base address is fixed at 0x00000000. The vector table contains the initialization value for the stack pointer on reset, and the entry point addresses for all exception handlers. The vector number on previous page defines the order of entries in the vector table associated with exception handler entry as illustrated in previous section.

Vector Table Word Offset	Description
0	SP_main – The Main stack pointer
Vector Number	Exception Entry Pointer using that Vector Number

Table 6-3 Vector Table Format

6.3.5 Operation Description

The NVIC interrupts can be enabled and disabled by writing to their corresponding Interrupt Set-Enable or Interrupt Clear-Enable register bit-field. The registers use a write-1-to-enable and write-1-to-clear policy, both registers reading back the current enabled state of the corresponding interrupts. When an interrupt is disabled, interrupt assertion will cause the interrupt to become Pending, however, the interrupt will not activate. If an interrupt is Active when it is disabled, it remains in its Active state until cleared by reset or an exception return. Clearing the enable bit prevents new activations of the associated interrupt.

The NVIC interrupts can be pended/un-pended using a complementary pair of registers to those used to enable/disable the interrupts, named the Set-Pending Register and Clear-Pending Register respectively. The registers use a write-1-to-enable and write-1-to-clear policy, both registers reading back the current pended state of the corresponding interrupts. The Clear-Pending Register has no effect on the execution status of an Active interrupt.

The NVIC interrupts are prioritized by updating an 8-bit field within a 32-bit register (each register supporting four interrupts).

The general registers associated with the NVIC are all accessible from a block of memory in the System Control Space and will be described in next section.

6.3.6 NVIC Control Register Map

R: read only**, W:** write only**, R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
SCS Base Address: **SCS_BA = 0xE000_E000**				
NVIC_ISER	SCS_BA+0x100	R/W	IRQ0~IRQ31 Set-Enable Control Register	0x0000_0000
NVIC_ICER	SCS_BA+0x180	R/W	IRQ0~IRQ31 Clear-Enable Control Register	0x0000_0000
NVIC_ISPR	SCS_BA+0x200	R/W	IRQ0~IRQ31 Set-Pending Control Register	0x0000_0000
NVIC_ICPR	SCS_BA+0x280	R/W	IRQ0~IRQ31Clear-Pending Control Register	0x0000_0000
NVIC_IPR0	SCS_BA+0x400	R/W	IRQ0~IRQ3 Priority Control Register	0x0000_0000
NVIC_IPR1	SCS_BA+0x404	R/W	IRQ4~IRQ7 Priority Control Register	0x0000_0000
NVIC_IPR2	SCS_BA+0x408	R/W	IRQ8~IRQ11 Priority Control Register	0x0000_0000
NVIC_IPR3	SCS_BA+0x40C	R/W	IRQ12~IRQ15 Priority Control Register	0x0000_0000
NVIC_IPR4	SCS_BA+0x410	R/W	IRQ16~IRQ19 Priority Control Register	0x0000_0000
NVIC_IPR5	SCS_BA+0x414	R/W	IRQ20~IRQ23 Priority Control Register	0x0000_0000
NVIC_IPR6	SCS_BA+0x418	R/W	IRQ24~IRQ27 Priority Control Register	0x0000_0000
NVIC_IPR7	SCS_BA+0x41C	R/W	IRQ28~IRQ31 Priority Control Register	0x0000_0000

6.3.7 NVIC Control Register Description

IRQ0 ~ IRQ31 Set-Enable Control Register （NVIC_ISER）

Register	Offset	R/W	Description	Reset Value
NVIC_ISER	SCS_BA+0x100	R/W	IRQ0~IRQ31 Set-Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SETENA							
23	22	21	20	19	18	17	16
SETENA							
15	14	13	12	11	10	9	8
SETENA							
7	6	5	4	3	2	1	0
SETENA							

Bits	Description	
[31:0]	SETENA	**Interrupt Enable Bits** Enable one or more interrupts. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47). Write Operation: 0 = No effect. 1 = Write 1 to enable associated interrupt. Read Operation: 0 = Associated interrupt status Disabled. 1 = Associated interrupt status Enabled. Read value indicates the current enable status.

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

IRQ0 ~ IRQ31 Clear-Enable Control Register (NVIC_ICER)

Register	Offset	R/W	Description	Reset Value
NVIC_ICER	SCS_BA+0x180	R/W	IRQ0~IRQ31 Clear-Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CLRENA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CLRENA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CLRENA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CLRENA							

Bits	Description	
[31:0]	**CLRENA**	**Interrupt Disable Bits** Disable one or more interrupts. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47). Write Operation: 0 = No effect. 1 = Write 1 to disable associated interrupt. Read Operation: 0 = Associated interrupt status Disabled. 1 = Associated interrupt status Enabled. Read value indicates the current enable status.

IRQ0 ~ IRQ31 Set-Pending Control Register （NVIC_ISPR）

Register	Offset	R/W	Description	Reset Value
NVIC_ISPR	SCS_BA+0x200	R/W	IRQ0~IRQ31 Set-Pending Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SETPEND							
23	22	21	20	19	18	17	16
SETPEND							
15	14	13	12	11	10	9	8
SETPEND							
7	6	5	4	3	2	1	0
SETPEND							

Bits	Description	
[31:0]	SETPEND	**Set Interrupt Pending** Write Operation: 0 = No effect. 1 = Write 1 to set pending state. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47). Read Operation: 0 = Associated interrupt in not in pending status. 1 = Associated interrupt is in pending status. Read value indicates the current pending status.

IRQ0 ~ IRQ31 Clear-Pending Control Register （NVIC_ICPR）

Register	Offset	R/W	Description	Reset Value
NVIC_ICPR	SCS_BA+0x280	R/W	IRQ0~IRQ31Clear-Pending Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CLRPEND							
23	22	21	20	19	18	17	16
CLRPEND							
15	14	13	12	11	10	9	8
CLRPEND							
7	6	5	4	3	2	1	0
CLRPEND							

Bits	Description	
[31:0]	**CLRPEND**	**Clear Interrupt Pending** Write Operation: 0 = No effect. 1 = Write 1 to clear pending state. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47). Read Operation: 0 = Associated interrupt in not in pending status. 1 = Associated interrupt is in pending status. Read value indicates the current pending status.

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

IRQ0 ~ IRQ3 Interrupt Priority Register （NVIC_IPR0）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR0	SCS_BA+0x400	R/W	IRQ0~IRQ3 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_3		Reserved					
23	22	21	20	19	18	17	16
PRI_2		Reserved					
15	14	13	12	11	10	9	8
PRI_1		Reserved					
7	6	5	4	3	2	1	0
PRI_0		Reserved					

Bits	Description	
[31:30]	PRI_3	**Priority of IRQ3** “0” denotes the highest priority and “3” denotes the lowest priority.
[23:22]	PRI_2	**Priority of IRQ2** “0” denotes the highest priority and “3” denotes the lowest priority.
[15:14]	PRI_1	**Priority of IRQ1** “0” denotes the highest priority and “3” denotes the lowest priority.
[7:6]	PRI_0	**Priority of IRQ0** “0” denotes the highest priority and “3” denotes the lowest priority.

IRQ4 ~ IRQ7 Interrupt Priority Register （NVIC_IPR1）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR1	SCS_BA+0x404	R/W	IRQ4~IRQ7 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_7		Reserved					
23	22	21	20	19	18	17	16
PRI_6		Reserved					
15	14	13	12	11	10	9	8
PRI_5		Reserved					
7	6	5	4	3	2	1	0
PRI_4		Reserved					

Bits	Description	
[31:30]	PRI_7	**Priority of IRQ7** “0” denotes the highest priority and “3” denotes the lowest priority.
[23:22]	PRI_6	**Priority of IRQ6** “0” denotes the highest priority and “3” denotes the lowest priority.
[15:14]	PRI_5	**Priority of IRQ5** “0” denotes the highest priority and “3” denotes the lowest priority.
[7:6]	PRI_4	**Priority of IRQ4** “0” denotes the highest priority and “3” denotes the lowest priority.

IRQ8 ~ IRQ11 Interrupt Priority Register （NVIC_IPR2）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR2	SCS_BA+0x408	R/W	IRQ8~IRQ11 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_11		Reserved					
23	22	21	20	19	18	17	16
PRI_10		Reserved					
15	14	13	12	11	10	9	8
PRI_9		Reserved					
7	6	5	4	3	2	1	0
PRI_8		Reserved					

Bits	Description	
[31:30]	PRI_11	**Priority of IRQ11** “0” denotes the highest priority and “3” denotes the lowest priority.
[23:22]	PRI_10	**Priority of IRQ10** “0” denotes the highest priority and “3” denotes the lowest priority.
[15:14]	PRI_9	**Priority of IRQ9** “0” denotes the highest priority and “3” denotes the lowest priority.
[7:6]	PRI_8	**Priority of IRQ8** “0” denotes the highest priority and “3” denotes the lowest priority.

IRQ12 ~ IRQ15 Interrupt Priority Register （NVIC_IPR3）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR3	SCS_BA+0x40C	R/W	IRQ12~IRQ15 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_15		Reserved					
23	22	21	20	19	18	17	16
PRI_14		Reserved					
15	14	13	12	11	10	9	8
PRI_13		Reserved					
7	6	5	4	3	2	1	0
PRI_12		Reserved					

Bits	Description	
[31:30]	PRI_15	**Priority of IRQ15** “0” denotes the highest priority and “3” denotes the lowest priority.
[23:22]	PRI_14	**Priority of IRQ14** “0” denotes the highest priority and “3” denotes the lowest priority.
[15:14]	PRI_13	**Priority of IRQ13** “0” denotes the highest priority and “3” denotes the lowest priority.
[7:6]	PRI_12	**Priority of IRQ12** “0” denotes the highest priority and “3” denotes the lowest priority.

IRQ16 ~ IRQ19 Interrupt Priority Register （NVIC_IPR4）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR4	SCS_BA+0x410	R/W	IRQ16~IRQ19 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_19		Reserved					
23	22	21	20	19	18	17	16
PRI_18		Reserved					
15	14	13	12	11	10	9	8
PRI_17		Reserved					
7	6	5	4	3	2	1	0
PRI_16		Reserved					

Bits	Description	
[31:30]	PRI_19	**Priority of IRQ19** “0” denotes the highest priority and “3” denotes the lowest priority.
[23:22]	PRI_18	**Priority of IRQ18** “0” denotes the highest priority and “3” denotes the lowest priority.
[15:14]	PRI_17	**Priority of IRQ17** “0” denotes the highest priority and “3” denotes the lowest priority.
[7:6]	PRI_16	**Priority of IRQ16** “0” denotes the highest priority and “3” denotes the lowest priority.

IRQ20 ~ IRQ23 Interrupt Priority Register （NVIC_IPR5）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR5	SCS_BA+0x414	R/W	IRQ20~IRQ23 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_23		Reserved					
23	22	21	20	19	18	17	16
PRI_22		Reserved					
15	14	13	12	11	10	9	8
PRI_21		Reserved					
7	6	5	4	3	2	1	0
PRI_20		Reserved					

Bits	Description	
[31:30]	PRI_23	**Priority of IRQ23** “0” denotes the highest priority and “3” denotes the lowest priority.
[23:22]	PRI_22	**Priority of IRQ22** “0” denotes the highest priority and “3” denotes the lowest priority.
[15:14]	PRI_21	**Priority of IRQ21** “0” denotes the highest priority and “3” denotes the lowest priority.
[7:6]	PRI_20	**Priority of IRQ20** “0” denotes the highest priority and “3” denotes the lowest priority.

IRQ24 ~ IRQ27 Interrupt Priority Register （NVIC_IPR6）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR6	SCS_BA+0x418	R/W	IRQ24~IRQ27 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_27		Reserved					
23	22	21	20	19	18	17	16
PRI_26		Reserved					
15	14	13	12	11	10	9	8
PRI_25		Reserved					
7	6	5	4	3	2	1	0
PRI_24		Reserved					

Bits	Description	
[31:30]	PRI_27	**Priority of IRQ27** "0" denotes the highest priority and "3" denotes the lowest priority.
[23:22]	PRI_26	**Priority of IRQ26** "0" denotes the highest priority and "3" denotes the lowest priority.
[15:14]	PRI_25	**Priority of IRQ25** "0" denotes the highest priority and "3" denotes the lowest priority.
[7:6]	PRI_24	**Priority of IRQ24** "0" denotes the highest priority and "3" denotes the lowest priority.

IRQ28 ~ IRQ31 Interrupt Priority Register （NVIC_IPR7）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR7	SCS_BA+0x41C	R/W	IRQ28~IRQ31 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_31		Reserved					
23	22	21	20	19	18	17	16
PRI_30		Reserved					
15	14	13	12	11	10	9	8
PRI_29		Reserved					
7	6	5	4	3	2	1	0
PRI_28		Reserved					

Bits	Description	
[31:30]	PRI_31	**Priority of IRQ31** “0” denotes the highest priority and “3” denotes the lowest priority.
[23:22]	PRI_30	**Priority of IRQ30** “0” denotes the highest priority and “3” denotes the lowest priority.
[15:14]	PRI_29	**Priority of IRQ29** “0” denotes the highest priority and “3” denotes the lowest priority.
[7:6]	PRI_28	**Priority of IRQ28** “0” denotes the highest priority and “3” denotes the lowest priority.

6.3.8 Interrupt Source Control Registers

Besides the interrupt control registers associated with the NVIC, the Nano112 serials also implement some specific control registers to facilitate the interrupt functions, including “interrupt source identify”, and ”NMI source selection”, which are described below.

6.3.8.1 Interrupt Source Control Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
SCS Base Address: **INT_BA = 0x5000_0300**				
IRQ0_SRC	INT_BA+0x00	R	MCU IRQ0 (BOD_INT) interrupt source identify	0xXXXX_XXXX
IRQ1_SRC	INT_BA+0x04	R	MCU IRQ1 (WDT_INT) interrupt source identify	0xXXXX_XXXX
IRQ2_SRC	INT_BA+0x08	R	MCU IRQ2 (EINT0) interrupt source identify	0xXXXX_XXXX
IRQ3_SRC	INT_BA+0x0C	R	MCU IRQ3 (EINT1) interrupt source identify	0xXXXX_XXXX
IRQ4_SRC	INT_BA+0x10	R	MCU IRQ4 (GPABC_INT) interrupt source identify	0xXXXX_XXXX
IRQ5_SRC	INT_BA+0x14	R	MCU IRQ5 (GPDEF_INT) interrupt source identify	0xXXXX_XXXX
IRQ6_SRC	INT_BA+0x18	R	MCU IRQ6 (PWM0_INT) interrupt source identify	0xXXXX_XXXX
IRQ7_SRC	INT_BA+0x1C	R	Reserved	0x0000_0000
IRQ8_SRC	INT_BA+0x20	R	MCU IRQ8 (TMR0_INT) interrupt source identify	0xXXXX_XXXX
IRQ9_SRC	INT_BA+0x24	R	MCU IRQ9 (TMR1_INT) interrupt source identify	0xXXXX_XXXX
IRQ10_SRC	INT_BA+0x28	R	MCU IRQ10 (TMR2_INT) interrupt source identify	0xXXXX_XXXX
IRQ11_SRC	INT_BA+0x2C	R	MCU IRQ11 (TMR3_INT) interrupt source identify	0xXXXX_XXXX
IRQ12_SRC	INT_BA+0x30	R	MCU IRQ12 (UART0_INT) interrupt source identify	0xXXXX_XXXX
IRQ13_SRC	INT_BA+0x34	R	MCU IRQ13 (UART1_INT) interrupt source identify	0xXXXX_XXXX
IRQ14_SRC	INT_BA+0x38	R	MCU IRQ14 (SPI0_INT) interrupt source identify	0xXXXX_XXXX
IRQ15_SRC	INT_BA+0x3C	R	MCU IRQ15 (SPI1_INT) interrupt source identify	0xXXXX_XXXX
IRQ16_SRC	INT_BA+0x40	R	Reserved	0x0000_0000
IRQ17_SRC	INT_BA+0x44	R	MCU IRQ17 (IRC_INT) interrupt source identify	0xXXXX_XXXX

IRQ18_SRC	INT_BA+0x48	R	MCU IRQ18 (I2C0_INT) interrupt source identify	0xXXXX_XXXX
IRQ19_SRC	INT_BA+0x4C	R	MCU IRQ19 (I2C1_INT) interrupt source identify	0xXXXX_XXXX
IRQ20_SRC	INT_BA+0x50	R	Reserved	0x0000_0000
IRQ21_SRC	INT_BA+0x54	R	MCU IRQ21 (SC0_INT) interrupt source identify	0xXXXX_XXXX
IRQ22_SRC	INT_BA+0x58	R	MCU IRQ22 (SC1_INT) interrupt source identify	0xXXXX_XXXX
IRQ23_SRC	INT_BA+0x5C	R	Reserved	0x0000_0000
IRQ24_SRC	INT_BA+0x60	R	MCU IRQ24 (CKSD_INT) interrupt source identify	0xXXXX_XXXX
IRQ25_SRC	INT_BA+0x64	R	MCU IRQ25 (LCD_INT) interrupt source identify	0xXXXX_XXXX
IRQ26_SRC	INT_BA+0x68	R	MCU IRQ26 (DMA_INT) interrupt source identify	0xXXXX_XXXX
IRQ27_SRC	INT_BA+0x6C	R	Reserved	0x0000_0000
IRQ28_SRC	INT_BA+0x70	R	MCU IRQ28 (PDWU_INT) interrupt source identify	0xXXXX_XXXX
IRQ29_SRC	INT_BA+0x74	R	MCU IRQ29 (ADC_INT) interrupt source identify	0xXXXX_XXXX
IRQ30_SRC	INT_BA+0x78	R	MCU IRQ30 (ACMP_INT) interrupt source identify	0xXXXX_XXXX
IRQ31_SRC	INT_BA+0x7C	R	MCU IRQ31 (RTC_INT) interrupt source identify	0xXXXX_XXXX
NMI_SEL	INT_BA+0x80	R/W	NMI Source Interrupt Select Control Register	0x0000_0000
MCU_IRQ	INT_BA+0x84	R/W	MCU Interrupt Request Source Register	0x0000_0000

6.3.8.2 Interrupt source control register description

Interrupt Source Identify Register (IRQn_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ0_SRC	INT_BA+0x00	R	MCU IRQ0 (BOD_INT) interrupt source identify	0xXXXX_XXXX
IRQ1_SRC	INT_BA+0x04	R	MCU IRQ1 (WDT_INT) interrupt source identify	0xXXXX_XXXX
IRQ2_SRC	INT_BA+0x08	R	MCU IRQ2 (EINT0) interrupt source identify	0xXXXX_XXXX
IRQ3_SRC	INT_BA+0x0C	R	MCU IRQ3 (EINT1) interrupt source identify	0xXXXX_XXXX
IRQ4_SRC	INT_BA+0x10	R	MCU IRQ4 (GPABC_INT) interrupt source identify	0xXXXX_XXXX
IRQ5_SRC	INT_BA+0x14	R	MCU IRQ5 (GPDEF_INT) interrupt source identify	0xXXXX_XXXX
IRQ6_SRC	INT_BA+0x18	R	MCU IRQ6 (PWM0_INT) interrupt source identify	0xXXXX_XXXX
IRQ7_SRC	INT_BA+0x1C	R	Reserved	0x0000_0000
IRQ8_SRC	INT_BA+0x20	R	MCU IRQ8 (TMR0_INT) interrupt source identify	0xXXXX_XXXX
IRQ9_SRC	INT_BA+0x24	R	MCU IRQ9 (TMR1_INT) interrupt source identify	0xXXXX_XXXX
IRQ10_SRC	INT_BA+0x28	R	MCU IRQ10 (TMR2_INT) interrupt source identify	0xXXXX_XXXX
IRQ11_SRC	INT_BA+0x2C	R	MCU IRQ11 (TMR3_INT) interrupt source identify	0xXXXX_XXXX
IRQ12_SRC	INT_BA+0x30	R	MCU IRQ12 (UART0_INT) interrupt source identify	0xXXXX_XXXX
IRQ13_SRC	INT_BA+0x34	R	MCU IRQ13 (UART1_INT) interrupt source identify	0xXXXX_XXXX
IRQ14_SRC	INT_BA+0x38	R	MCU IRQ14 (SPI0_INT) interrupt source identify	0xXXXX_XXXX
IRQ15_SRC	INT_BA+0x3C	R	MCU IRQ15 (SPI1_INT) interrupt source identify	0xXXXX_XXXX
IRQ16_SRC	INT_BA+0x40	R	Reserved	0x0000_0000
IRQ17_SRC	INT_BA+0x44	R	MCU IRQ17 (IRC_INT) interrupt source identify	0xXXXX_XXXX
IRQ18_SRC	INT_BA+0x48	R	MCU IRQ18 (I2C0_INT) interrupt source identify	0xXXXX_XXXX
IRQ19_SRC	INT_BA+0x4C	R	MCU IRQ19 (I2C1_INT) interrupt source identify	0xXXXX_XXXX
IRQ20_SRC	INT_BA+0x50	R	Reserved	0x0000_0000
IRQ21_SRC	INT_BA+0x54	R	MCU IRQ21 (SC0_INT) interrupt source identify	0xXXXX_XXXX
IRQ22_SRC	INT_BA+0x58	R	MCU IRQ22 (SC1_INT) interrupt source identify	0xXXXX_XXXX

IRQ23_SRC	INT_BA+0x5C	R	Reserved	0x0000_0000
IRQ24_SRC	INT_BA+0x60	R	MCU IRQ24 (CKSD_INT) interrupt source identify	0xXXXX_XXXX
IRQ25_SRC	INT_BA+0x64	R	MCU IRQ25 (LCD_INT) interrupt source identify	0xXXXX_XXXX
IRQ26_SRC	INT_BA+0x68	R	MCU IRQ26 (DMA_INT) interrupt source identify	0xXXXX_XXXX
IRQ27_SRC	INT_BA+0x6C	R	Reserved	0x0000_0000
IRQ28_SRC	INT_BA+0x70	R	MCU IRQ28 (PDWU_INT) interrupt source identify	0xXXXX_XXXX
IRQ29_SRC	INT_BA+0x74	R	MCU IRQ29 (ADC_INT) interrupt source identify	0xXXXX_XXXX
IRQ30_SRC	INT_BA+0x78	R	MCU IRQ30 (ACMP_INT) interrupt source identify	0xXXXX_XXXX
IRQ31_SRC	INT_BA+0x7C	R	MCU IRQ31 (RTC_INT) interrupt source identify	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**INT_SRC**			

Bits	**Description**	
[31:4]	**Reserved**	Reserved.
[3:0]	**INT_SRC**	**Interrupt Source** Define the interrupt sources for interrupt event.

Address	**INT Number**	**Bits**	**Description**

INT_BA+0x00	0	[2:0]	Bit2 = 1'b0
			Bit1 = 1'b0
			Bit0 = BOD_INT

INT_BA+0x04	1	[2:0]	Bit2 = 1'b0
			Bit1 = 1'b0
			Bit0 = WDT_INT

INT_BA+0x08	2	[2:0]	Bit2 = 1'b0
			Bit1 = 1'b0

Address	INT Number	Bits	Description
			Bit0 = EINT0 – external interrupt 0 from PB.14

INT_BA+0x0C	3	[2:0]	Bit2 = 1’b0
			Bit1 = 1’b0
			Bit0 = EINT1 – external interrupt 1 from PB.15

INT_BA+0x10	4	[2:0]	Bit2 = GPC_INT
			Bit1 = GPB_INT
			Bit0 = GPA_INT

INT_BA+0x14	5	[2:0]	Bit2 = GPF_INT
			Bit1 = GPE_INT
			Bit0 = GPD_INT

INT_BA+0x18	6	[3:0]	Bit3 = PWM0_CH3_INT
			Bit2 = PWM0_CH2_INT
			Bit1 = PWM0_CH1_INT
			Bit0 = PWM0_CH0_INT

INT_BA+0x1C	7	[3:0]	Reserved

INT_BA+0x20	8	[2:0]	Bit2 = 1’b0
			Bit1 = 1’b0
			Bit0 = TMR0_INT

INT_BA+0x24	9	[2:0]	Bit2 = 1’b0
			Bit1 = 1’b0
			Bit0 = TMR1_INT

INT_BA+0x28	10	[2:0]	Bit2 = 1’b0
			Bit1 = 1’b0
			Bit0 = TMR2_INT

INT_BA+0x2C	11	[2:0]	Bit2 = 1’b0
			Bit1 = 1’b0
			Bit0 = TMR3_INT

INT_BA+0x30	12	[2:0]	Bit2 = 1’b0
			Bit1 = 1’b0
			Bit0 = UART0_INT

INT_BA+0x34	13	[2:0]	Bit2 = 1’b0
			Bit1 = 1’b0
			Bit0 = UART1_INT

INT_BA+0x38	14	[2:0]	Bit2 = 1’b0
			Bit1 = 1’b0
			Bit0 = SPI0_INT

INT_BA+0x3C	15	[2:0]	Bit2 = 1’b0
			Bit1 = 1’b0
			Bit0 = SPI1_INT

INT_BA+0x40	16	[2:0]	Reserved

INT_BA+0x44	17	[2:0]	Bit2 = 1’b0
			Bit1 = 1’b0

Address	INT Number	Bits	Description
			Bit0 = IRC_INT

INT_BA+0x48	18	[2:0]	Bit2 = 1'b0
			Bit1 = 1'b0
			Bit0 = I2C0_INT

INT_BA+0x4C	19	[2:0]	Bit2 = 1'b0
			Bit1 = 1'b0
			Bit0 = I2C1_INT

INT_BA+0x50	20	[2:0]	Reserved

INT_BA+0x54	21	[2:0]	Bit2 = 1'b0
			Bit1 = 1'b0
			Bit0 = SC0_INT

INT_BA+0x58	22	[2:0]	Bit2 = 1'b0
			Bit1 = 1'b0
			Bit0 = SC1_INT

INT_BA+0x5C	23	[2:0]	Reserved
INT_BA+0x60	24	[2:0]	Reserved

INT_BA+0x64	25	[2:0]	Bit2 = 1'b0
			Bit1 = 1'b0
			Bit0 = LCD_INT

INT_BA+0x68	26	[2:0]	Bit2 = 1'b0
			Bit1 = 1'b0
			Bit0 = DMA_INT

INT_BA+0x6C	27	[2:0]	Reserved

INT_BA+0x70	28	[2:0]	Bit2 = 1'b0
			Bit1 = 1'b0
			Bit0 = PD_WU_INT

INT_BA+0x74	29	[2:0]	Bit2 = 1'b0
			Bit1 = 1'b0
			Bit0 = ADC_INT

INT_BA+0x78	30	[2:0]	Reserved

INT_BA+0x7C	31	[2:0]	Bit2 = 1'b0
			Bit1 = 1'b0
			Bit0 = RTC_INT

NMI Interrupt Source Select Control Register (NMI_SEL)

Register	Offset	R/W	Description	Reset Value
NMI_SEL	INT_BA+0x80	R/W	NMI Source Interrupt Select Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**NMI_SEL**				

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4:0]	**NMI_SEL**	**The NMI Interrupt to Cortex-m0 Can Be Selected from One of the Interrupt[31:0]** The NMI_SEL bit[4:0] is used to select the NMI interrupt source

MCU Interrupt Request Source Register (MCU_IRQ)

Register	Offset	R/W	Description	Reset Value
MCU_IRQ	INT_BA+0x84	R/W	MCU Interrupt Request Source Register	0x0000_0000

31	30	29	28	27	26	25	24
MCU_IRQ							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
MCU_IRQ							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MCU_IRQ							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MCU_IRQ							

Bits	Description	
[31:0]	**MCU_IRQ**	**MCU IRQ Source Bits** The MCU_IRQ collects all the interrupts from the peripherals and generates the synchronous interrupt to MCU Cortex-M0. There are two modes to generate interrupt to Cortex-M0, the normal mode. The MCU_IRQ collects all interrupts from each peripheral and synchronizes them and then interrupts the Cortex-M0. When the MCU_IRQ[n] is “0”, setting MCU_IRQ[n] “1” will generate an interrupt to Cortex_M0 NVIC[n]. When the MCU_IRQ[n] is “1” (means an interrupt is asserted), setting the MCU_bit[n] will clear the interrupt Set MCU_IRQ[n] “0”: no any effect

6.4 System Manager

6.4.1 Overview

System manager mainly controls the power modes, wake-up sources, power architecture, reset sources, scalable LDO and system memory map. It also provides information about product ID and multi-function pin control.

6.4.2 Features

- Power modes and wake-up sources
- System power architecture
- Reset sources
- Scalable LDO
- System Memory Map
- System manager registers for:
  - Part Number ID
  - Multi-functional pin control

6.4.3 Functional Description

6.4.3.1 Power modes and Wake-up sources

There are three power modes. There are several wake-up sources in Idle mode and Power-down mode. The available clocks depend on each power mode.

Power Modes:

- Normal mode:
  - Definition: CPU is active.
  - Entry condition: chip is in this power after system reset released
  - Wake-up sources: N/A
  - Available clocks: All.
  - After wake-up: N/A
- Idle mode:
  - Definition: CPU is in sleep mode.
  - Entry condition: CPU sets sleep mode enable and executes WFI instruction.
  - Wake-up source: all interrupts
  - Available clocks: all except CPU clock.
  - After wake-up: CPU back to normal mode.
- Power-down mode:
  - Definition: CPU is in sleep mode and all clocks stop except LXT and LIRC. SRAM content is retained.

- Entry condition: CPU sets sleep mode enable and power down enable and executes WFI instruction.
- Wake-up sources: RTC, WDT, I^{2}C, Timer, UART, SPI, BOD, GPIO
- Available clocks: LXT and LIRC
- After wake-up: CPU back to normal mode

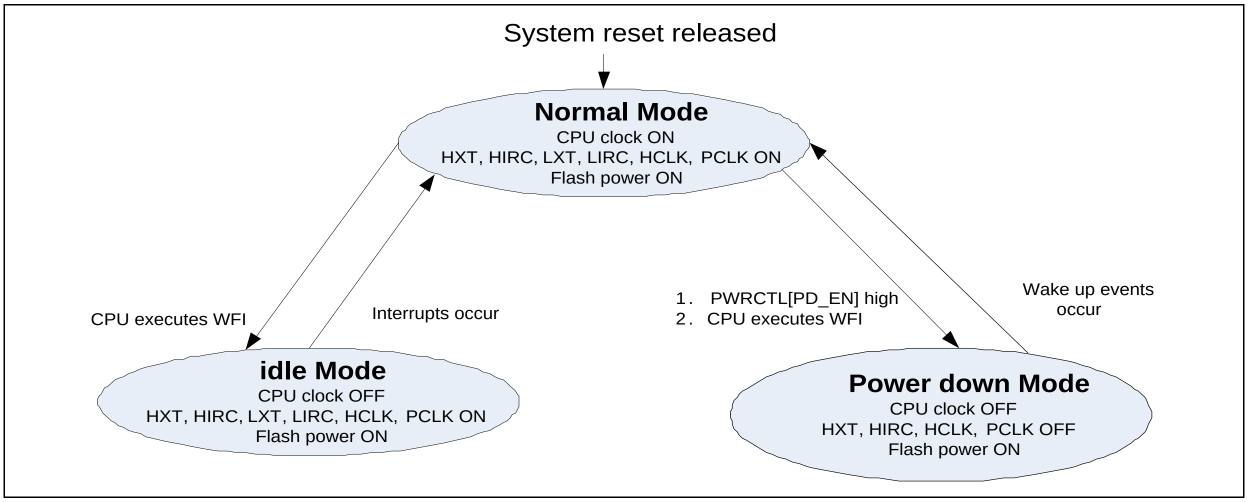


Figure 6‑2 Power Modes

	Normal Mode	Idle Mode	Power-Down Mode
Wake-up time to Normal mode	0	0	6us: from wake-up event to first CPU core valid clock
HXT (4~24 MHz XTL)	ON	ON	Halt
HIRC (12/16 MHz OSC)	ON	ON	Halt
LXT (32768 Hz XTL)	ON	ON	ON/OFF[1]
LIRC (10 kHz OSC)	ON	ON	ON/OFF[2]
PLL	ON	ON	Halt
LDO	ON	ON	ON
CPU	ON	Halt	Halt
HCLK/PCLK	ON	ON	Halt
SRAM retention	ON	ON	ON
FLASH	ON	ON	Halt
GPIO	ON	ON	Halt
DMA	ON	ON	Halt
I^{2}C	ON	ON	Halt
PWM	ON	ON	Halt
TIMER	ON	ON	Halt

UART	ON	ON	Halt
SPI	ON	ON	Halt
RTC	ON	ON	ON/OFF[3]
WDT	ON	ON	ON/OFF[4]
LCD	ON	ON	ON/OFF[5]
ADC	ON	ON	Halt

Table 6‒4 Clocks in Power Modes

1. LXT (32768 Hz XTL) ON or OFF is depended on SW setting in run mode
2. LIRC (10 kHz OSC) ON or OFF is depended on S/W setting in run mode
3. RTC functions if LXT is ON
4. WDT functions if LIRC is ON.
5. LCD functions if LXT is ON.

Mode Name	Entry	Wake-up Sources	Available Clocks	LDO
Normal	After power release	NA	all	ON
HXT (4~24 MHz XTL)	ON	ON	Halt	
HIRC (12/16 MHz OSC)	ON	ON	Halt	
LXT (32768 Hz XTL)	ON	ON	ON/OFF[1]	
LIRC (10 kHz OSC)	ON	ON	ON/OFF[2]	
PLL	ON	ON	Halt	
LDO	ON	ON	ON	
CPU	ON	Halt	Halt	
HCLK/PCLK	ON	ON	Halt	
SRAM retention	ON	ON	ON	
FLASH	ON	ON	Halt	
GPIO	ON	ON	Halt	
DMA	ON	ON	Halt	
I^{2}C	ON	ON	Halt	

Table 6‒5 Summary of Power Modes

Wake-up sources in Power-down mode:

RTC, WDT, I²C, Timer, UART, SPI, BOD, GPIO

After chip enters power down, the following wake-up sources can wake chip up to normal mode and list the condition about how to enter Power-down mode again for each peripheral.

Wake-Up Source	Wake-Up Condition	System Can Enter Power-Down Mode Again Condition (User Need To Wait This Condition Before Setting PD_EN(PWRCTL[6]) And Execute WFI To Enter Power-down mode)
External	-	After user write 1 to clear the GPIOx_ISRC bit.
Interrupts		
UART	Data wake-up	None
	CTSn wake-up	Requiring 2 UART_CLK after wake-up.
GPIO	-	After user write 1 to clear the GPIOx_ISRC bit.
RTC	-	Requiring 1 RTC_CLK (about 30 µs) after wake-up.
SPI	-	After SPI slave clock goes from high to low (depending on exteral SPI master)
Timer	TMRx_ISR[TCAP_IS]	After software sets timer TMRx_ISR[SW_RST] (software reset) or software writes 1 to clear TMRx_ISR[TCAP_IS].
	TMRx_ISR[TMR_IS]	After software sets timer TMRx_ISR[SW_RST] (software reset) or software writes 1 to clear TMRx_ISR[TMR_IS].
WDT	-	None
BOD	-	After voltage is raised higher than target voltage or BODx_INT_EN is set to low.
I²C	-	None

Table 6−6 Condition of Entering Power-down Mode Again Table

6.4.3.2 System Resets

The system reset includes the events listed below. These reset events can be read by the RST_SRC register.

- Hardware Reset
  - Power-on Reset (POR)
  - Brown-out Reset (BOD)
  - Low level on the nRESET pin
  - Watchdog Timer Time-out Reset
- Software Reset
  - MCU Reset – SYSRESETREQ(AIRCR[2]
  - Cortex-M0 MCU One-shot Reset- CPU_RST(IPRSTC1[1])
  - Chip One-shot Reset – CHIP_RST(IPRSTC1[0])

6.4.3.3 System Power Architecture

In this chip, the power distribution is divided into two segments.

- Analog power from AV_{DD} and AV_{SS} provides the power for analog components operation. AV_{DD}

must be equal to V_{DD} to avoid leakage current.

- Digital power from V_{DD} and V_{SS} supplies power to the I/O pins and internal regulator which provides a scalable1.8V/1.6V power for digital operation.

The internal voltage regulator needs an external capacitor (LDO_CAP) to provide stable power for digital operation. Analog power (AV_{DD}) should be the same voltage level as the digital power (V_{DD}). The following figure shows the power distribution.

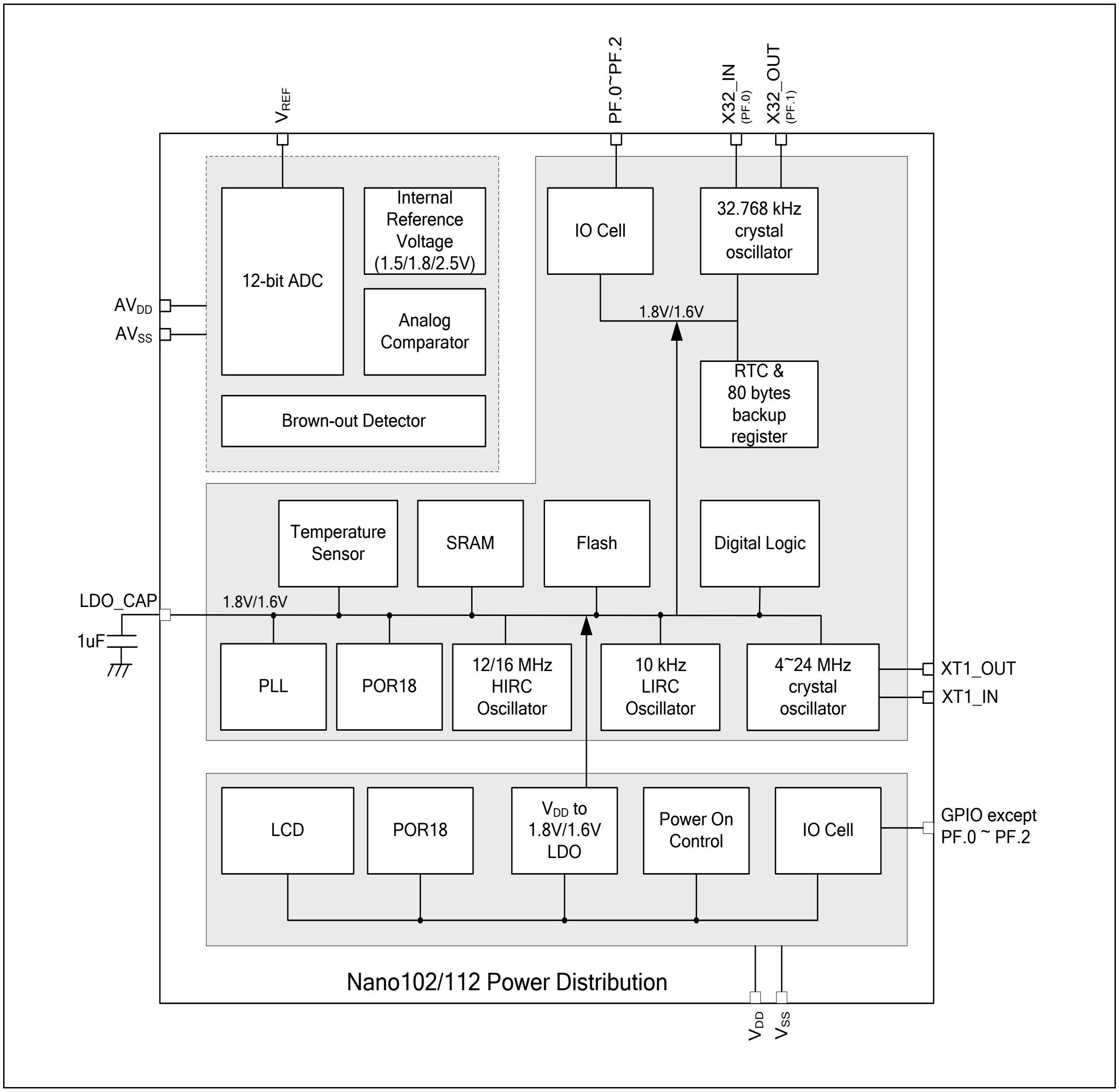


Figure 6-3 Power Architecture

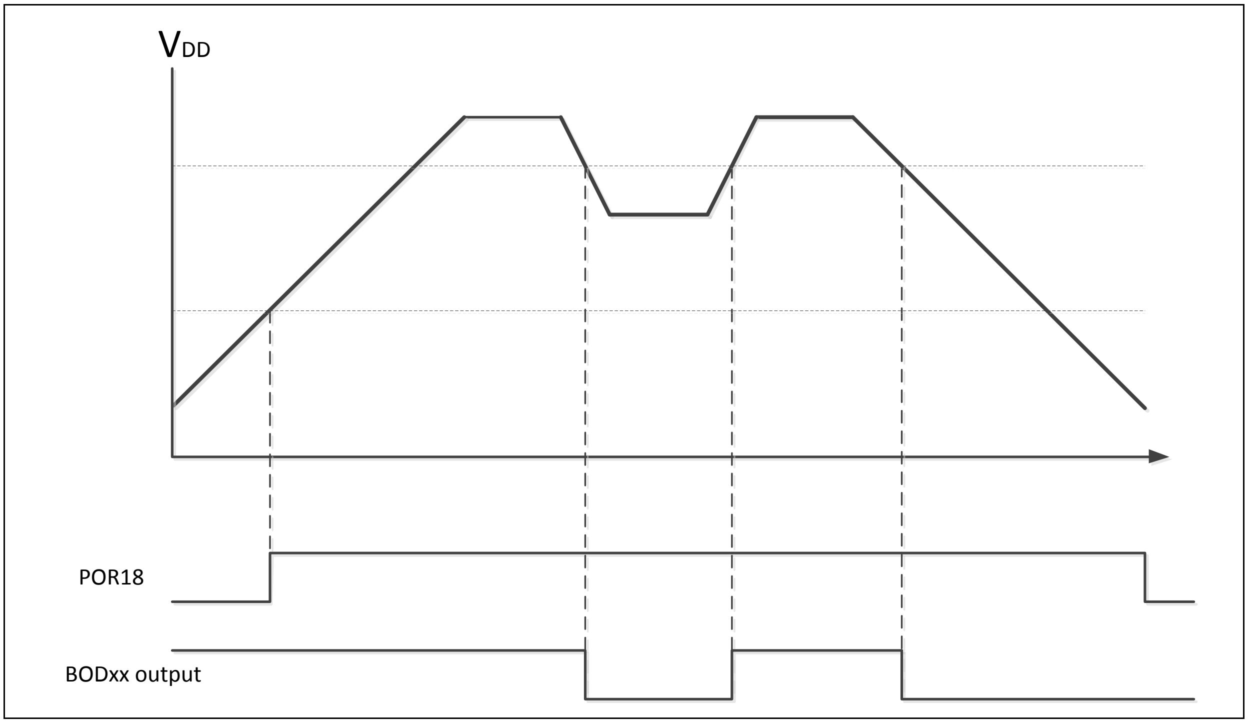


Figure 6-4 Power Supervisors

6.4.3.4 *Scalable LDO*

This chip provides scalable LDO to balance the performance and power consumption. There are two options of LDO output voltage: 1.8V and 1.6V. Use 1.8V option when higher performance (operating speed higher than 12MHz) is required. Use 1.6V option when lower power is demand.

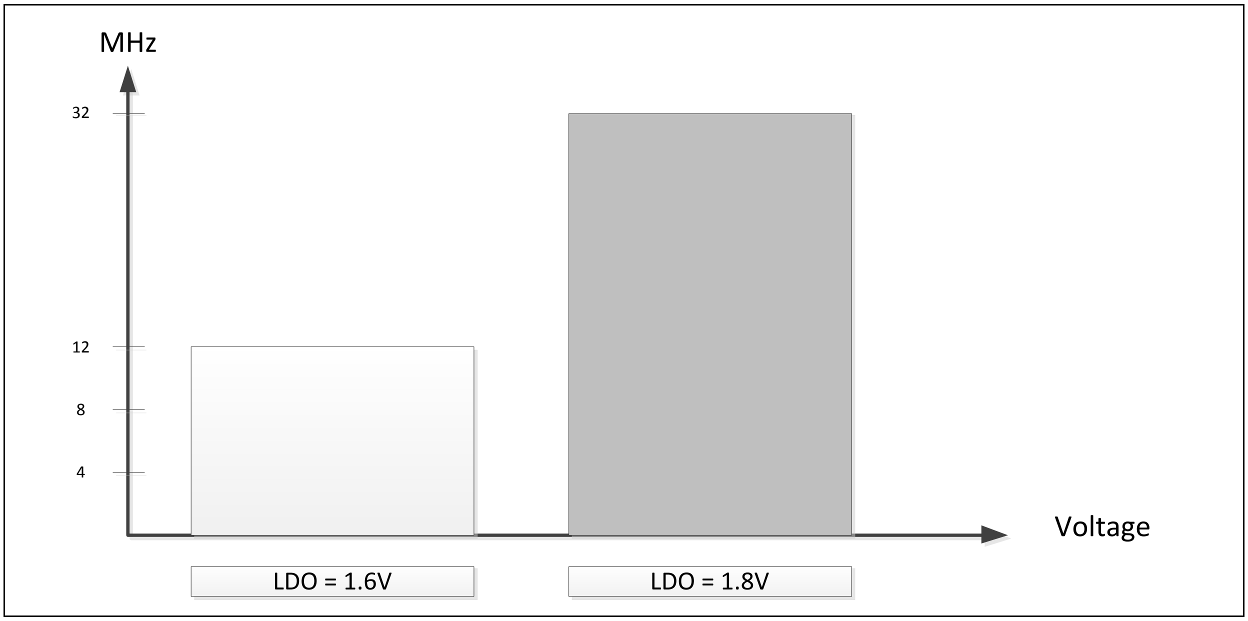


Figure 6-5 Scalable LDO and Performance

6.4.3.5 Auto-trim

This chip supports auto-trim function: the HIRC trim (12/16 MHz RC oscillator), according to the accurate LXT (32.768 kHz crystal oscillator), automatically gets accurate HIRC output frequency, 1 % deviation within all temperature ranges. For instance, PLL needs an accurate 12 MHz input clock to output accurate clock. In such case, if users do not want to use 12 MHz HXT as PLL input clock, they need to solder 32.768 kHz crystal in system, and set the HIRC target output clock to 12 MHz by setting TRIM_SEL (IRCTRIMCTL[1:0] trim frequency selection) to “10”, and the auto-trim function will be enabled. Interrupt status bit FREQ_LOCK (IRCTRIMINT[0] HIRC frequency lock status) high indicates the HIRC output frequency is accurate within 1% deviation. To get better results, it is recommended to set both TRIM_LOOP (IRCTRIMCTL[5:4] trim calculation loop) and TRIM_RETRY_CNT (IRCTRIMCTL[7:6] trim value update limitation count) to “11”.

6.4.4 Register Map

Open lock sequence for protected registers

Some of the system control registers need to be protected to avoid inadvertent write and disturb the chip operation. These system control registers are locked after the power-on reset till users to open the lock. For users to program these protected registers, an open lock sequence needs to be followed by a special programming sequence. The open sequence is to continually write the data “59h”, “16h” “88h” to the key controller address 0x5000_0100(RegLockAddr). Any different data value or different sequence or any other write operations to any other address during these three data program aborts the whole sequence. Therefore, users only follow the order of these three data and do not need to care the time interval when writing them.

After the lock is opened, users can check the lock bit RegLockAddr[0]. “1” is unlocked, “0” is locked. Then users can update the target register value if RegUnLock is high and write any data to the address “0x5000_0100” to re-lock the protected registers

Register	Offset	R/W	Description	Reset Value
GCR Base Address: **GCR_BA = 0x5000_0000**				
PDID	GCR_BA+0x00	R	Part Device Identification Number Register	0x0011_XXXX
RST_SRC	GCR_BA+0x04	R/W	System Reset Source Register	0x0000_00xx
IPRST_CTL1	GCR_BA+0x08	R/W	Peripheral Reset Control Resister1	0x0000_0000
IPRST_CTL2	GCR_BA+0x0C	R/W	Peripheral Reset Control Resister2	0x0000_0000
TEMPCTL	GCR_BA+0x20	R/W	Temperature Sensor Control Register	0x0000_0000
PA_L_MFP	GCR_BA+0x30	R/W	Port A Low Byte Multiple Function Control Register	0x0000_0000
PA_H_MFP	GCR_BA+0x34	R/W	Port A High Byte Multiple Function Control Register	0x0000_0000
PB_L_MFP	GCR_BA+0x38	R/W	Port B Low Byte Multiple Function Control Register	0x0000_0000
PB_H_MFP	GCR_BA+0x3C	R/W	Port B High Byte Multiple Function Control Register	0x0000_0000
PC_L_MFP	GCR_BA+0x40	R/W	Port C Low Byte Multiple Function Control Register	0x0000_0000
PC_H_MFP	GCR_BA+0x44	R/W	Port C High Byte Multiple Function Control Register	0x0000_0000
PD_L_MFP	GCR_BA+0x48	R/W	Port D Low Byte Multiple Function Control Register	0x0000_0000
PD_H_MFP	GCR_BA+0x4C	R/W	Port D High Byte Multiple Function Control Register	0x0000_0000
PE_L_MFP	GCR_BA+0x50	R/W	Port E Low Byte Multiple Function Control Register	0x0000_0000
PE_H_MFP	GCR_BA+0x54	R/W	Port E High Byte Multiple Function Control Register	0x0000_0000
PF_L_MFP	GCR_BA+0x58	R/W	Port F Low Byte Multiple Function Control Register	0x00FF_FFFF
PORCTL	GCR_BA+0x60	R/W	Power-On-Reset Controller Register	0x0000_0000
BODCTL	GCR_BA+0x64	R/W	Brown-out Detector Controller Register	0x00xx_x0xx

BODSTS	GCR_BA+0x68	R	Brown-out Detector Status Register	0x0000_0000
Int_VREFCTL	GCR_BA+0x6C	R/W	Internal Voltage Reference Control Register	0x0000_0700
LDO_CTL	GCR_BA+0x70	R/W	LDO Control Register	0x0000_0X0C
IRCTRIMCTL	GCR_BA+0x80	R/W	HIRC Trim Control Register	0x0000_0000
IRCTRIMIEN	GCR_BA+0x84	R/W	HIRC Trim Interrupt Enable Register	0x0000_0000
IRCTRIMINT	GCR_BA+0x88	R/W	HIRC Trim Interrupt Status Register	0x0000_0000
RegLockAddr	GCR_BA+0x100	R/W	Register Lock Key address	0x0000_0000

6.4.5 Register Description

Part Device Identification number Register (PDID)

Register	Offset	R/W	Description	Reset Value
PDID	GCR_BA+0x00	R	Part Device Identification Number Register	0x0011_XXXX

[1] Every part device identification has a unique default reset value.

31	30	29	28	27	26	25	24
PDID							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PDID							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PDID							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDID							

Bits	Description	
[31:0]	**PDID**	**Part Device ID** This register reflects device part number code. Software can read this register to identify which device is used.

NANO112 Series	Part Device ID
NANO112LB0AN	**0x00111201**
NANO112LB1AN	**0x00111202**
NANO112LC1AN	**0x00111203**
NANO112LC2AN	**0x00111204**
NANO112SB0AN	**0x00111205**
NANO112SB1AN	**0x00111206**
NANO112SC1AN	**0x00111207**
NANO112SC2AN	**0x00111208**
NANO112RB0AN	**0x00111209**
NANO112RB1AN	**0x00111210**
NANO112RC1AN	**0x00111211**
NANO112RC2AN	**0x00111212**
NANO112VB0AN	**0x00111213**
NANO112VB1AN	**0x00111214**
NANO112VC1AN	**0x00111214**

NANO112VC2AN	**0x00111216**
NANO102ZB0AN	**0x00110201**
NANO102ZB1AN	**0x00110202**
NANO102ZC1AN	**0x00110203**
NANO102ZC2ZN	**0x00110204**
NANO102LB0AN	**0x00110205**
NANO102LB1AN	**0x00110206**
NANO102LC1AN	**0x00110207**
NANO102LC2AN	**0x00110208**
NANO102SB0AN	**0x00110209**
NANO102SB1AN	**0x00110210**
NANO102SC1AN	**0x00110211**
NANO102SC2AN	**0x00110212**

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

System Reset Source Register (RST_SRC)

This register provides specific information for software to identify the chip’s reset source from the last operation.

Register	Offset	R/W	Description	Reset Value
RST_SRC	GCR_BA+0x04	R/W	System Reset Source Register	0x0000_00xx

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
RSTS_CPU	Reserved	RSTS_SYS	RSTS_BOD	Reserved	RSTS_WDT	RSTS_PAD	RSTS_POR

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**RSTS_CPU**	**The RSTS_CPU Flag Is Set by Hardware If Software Writes CPU_RST (IPRST_CTL1[1]) “1” to Rest Cortex-m0 Core and Flash Memory Controller (FMC)** 0 = No reset from CPU. 1 = Cortex-M0 core and FMC are reset by software setting CPU_RST to 1. **Note:** This bit is cleared by writing 1 to it.
[6]	**Reserved**	Reserved.
[5]	**RSTS_SYS**	**The RSTS_SYS Flag Is Set by the “Reset Signal” From the Cortex_M0 Kernel to Indicate the Previous Reset Source** 0 = No reset from Cortex_M0. 1 = Cortex_M0 had issued the reset signal to reset the system by writing 1 to the bit SYSRESTREQ(AIRCR[2], Application Interrupt and Reset Control Register) in system control registers of Cortex_M0 kernel. **Note:** This bit is cleared by writing 1 to it.
[4]	**RSTS_BOD**	**The RSTS_BOD Flag Is Set by the “Reset Signal” From the Brown-out-detected Module to Indicate the Previous Reset Source** 0 = No reset from BOD. 1 = Brown-out-Detected module had issued the reset signal to reset the system. **Note:** This bit is cleared by writing 1 to it.
[3]	**Reserved**	Reserved.
[2]	**RSTS_WDT**	**The RSTS_WDT Flag Is Set by the “Reset Signal” From the Watchdog Timer Module to Indicate the Previous Reset Source** 0 = No reset from Watchdog Timer. 1 = The Watchdog Timer module had issued the reset signal to reset the system. **Note:** This bit is cleared by writing 1 to it.
[1]	**RSTS_PAD**	**The RSTS_PAD Flag Is Set by the “Reset Signal” From the /RESET Pin or Power**

Bits	Description	
		Related Reset Sources to Indicate the Previous Reset Source 0 = No reset from nRESET pin. 1 = The /RESET pin had issued the reset signal to reset the system. **Note:** This bit is cleared by writing 1 to it.
[0]	**RSTS_POR**	**The RSTS_POR Flag Is Set by the “Reset Signal” From the Power-on Reset (POR) Module or Bit CHIP_RST (IPRSTC1[0]) to Indicate the Previous Reset Source** 0 = No reset from POR or CHIP_RST. 1 = Power-on Reset (POR) or CHIP_RST had issued the reset signal to reset the system. **Note:** This bit is cleared by writing 1 to it.

Peripheral Reset Control Register1 (IPRST_CTL1)

Register	Offset	R/W	Description	Reset Value
IPRST_CTL1	GCR_BA+0x08	R/W	Peripheral Reset Control Resister1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**DMA_RST**	**CPU_RST**	**CHIP_RST**

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**DMA_RST**	**DMA Controller Reset** This is a protected register. Please refer to open lock sequence to program it. Set this bit "1" will generate a reset signal to the DMA. SW needs to set this bit to low to release reset signal. 0 = Normal operation. 1 = DMA IP reset.
[1]	**CPU_RST**	**Cortex-m0 Core One-shot Reset** This is a protected register. Please refer to open lock sequence to program it. Setting this bit will only reset the Cortex-M0 core and Flash Memory Controller (FMC), and this bit will automatically return to "0" after the 2 clock cycles 0 = Normal. 1 = Reset Cortex-M0 core.
[0]	**CHIP_RST**	**Chip One-shot Reset** This is a protected register. Please refer to open lock sequence to program it. Setting this bit will reset the whole chip, including Cortex-M0 core and all peripherals like power-on reset and this bit will automatically return to "0" after the 2 clock cycles. The chip setting from flash will be also reloaded when chip one shot reset. 0 = Normal. 1 = Reset chip. **Note:** In the following conditions, chip setting from flash will be reloaded. Power-on Reset Brown-out-Detected Reset Low level on the nRESET pin Set IPRST_CTL1[CHIP_RST]

Peripheral Reset Control Register2 (IPRST_CTL2)

Register	Offset	R/W	Description	Reset Value
IPRST_CTL2	GCR_BA+0x0C	R/W	Peripheral Reset Control Resister2	0x0000_0000

31	30	29	28	27	26	25	24
SC1_RST	**SC0_RST**	**Reserved**	**ADC_RST**	**Reserved**	**LCD_RST**	**Reserved**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	**ACMP01_RST**	**Reserved**	**PWM0_RST**	**Reserved**		**UART1_RST**	**UART0_RST**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**SPI1_RST**	**SPI0_RST**	**Reserved**		**I2C1_RST**	**I2C0_RST**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**TMR3_RST**	**TMR2_RST**	**TMR1_RST**	**TMR0_RST**	**GPIO_RST**	**Reserved**

Bits	Description	
[31]	**SC1_RST**	**SmartCard1 Controller Reset** 0 = SmartCard module normal operation. 1 = SmartCard module reset.
[30]	**SC0_RST**	**SmartCard 0 Controller Reset** 0 = SmartCard module normal operation. 1 = SmartCard module reset.
[29]	**Reserved**	Reserved.
[28]	**ADC_RST**	**ADC Controller Reset** 0 = ADC module normal operation. 1 = ADC module reset.
[27]	**Reserved**	Reserved.
[26]	**LCD_RST**	**LCD Controller Reset** 0 = LCD module normal operation. 1 = LCD module reset.
[25:23]	**Reserved**	Reserved.
[22]	**ACMP01_RST**	**Comparator Controller Reset** 0 = Comparator module normal operation. 1 = Comparator module reset.
[21]	**Reserved**	Reserved.
[20]	**PWM0_RST**	**PWM0 Controller Reset** 0 = PWM0 module normal operation. 1 = PWM0 module reset.
[19:18]	**Reserved**	Reserved.
[17]	**UART1_RST**	**UART1 Controller Reset**

Bits	Description	
		0 = UART1 module normal operation. 1 = UART1 module reset.
[16]	**UART0_RST**	**UART0 Controller Reset** 0 = UART0 module normal operation. 1 = UART0 module reset.
[15:14]	**Reserved**	Reserved.
[13]	**SPI1_RST**	**SPI1 Controller Reset** 0 = SPI1 module normal operation. 1 = SPI1 module reset.
[12]	**SPI0_RST**	**SPI0 Controller Reset** 0 = SPI0 module normal operation. 1 = SPI0 module reset.
[11:10]	**Reserved**	Reserved.
[9]	**I2C1_RST**	**I2C1 Controller Reset** 0 = I2C1 module normal operation. 1 = I2C1 module reset.
[8]	**I2C0_RST**	**I2C0 Controller Reset** 0 = I2C0 module normal operation. 1 = I2C0 module reset.
[7:6]	**Reserved**	Reserved.
[5]	**TMR3_RST**	**Timer3 Controller Reset** 0 = Timer3 module normal operation. 1 = Timer3 module reset.
[4]	**TMR2_RST**	**Timer2 Controller Reset** 0 = Timer2 module normal operation. 1 = Timer2 module reset.
[3]	**TMR1_RST**	**Timer1 Controller Reset** 0 = Timer1 module normal operation. 1 = Timer1 module reset.
[2]	**TMR0_RST**	**Timer0 Controller Reset** 0 = Timer0 module normal operation. 1 = Timer0 module reset.
[1]	**GPIO_RST**	**GPIO Controller Reset** 0 = GPIO module normal operation. 1 = GPIO module reset.
[0]	**Reserved**	Reserved.

Temperature Sensor Control Register (TEMPCTL)

Register	Offset	R/W	Description	Reset Value
TEMPCTL	GCR_BA+0x20	R/W	Temperature Sensor Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**VTEMP_EN**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**VTEMP_EN**	**Temperature Sensor Enable Control** 0 = Temperature sensor function Disabled (default). 1 = Temperature sensor function Enabled.

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

Multiple Function Port A low byte Control Register (PA_L_MFP)

Register	Offset	R/W	Description	Reset Value
PA_L_MFP	GCR_BA+0x30	R/W	Port A Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PA7_MFP				PA6_MFP			
23	22	21	20	19	18	17	16
PA5_MFP				PA4_MFP			
15	14	13	12	11	10	9	8
PA3_MFP				PA2_MFP			
7	6	5	4	3	2	1	0
PA1_MFP				PA0_MFP			

Bits	Description		
[31:28]	PA7_MFP	PA.7 Pin Function Selection	
		PA7_MFP	Function
		0000	GPIOA[7]
		0010	ADC input channel 7
		0100	SmartCard1 card detect
[27:24]	PA6_MFP	PA.6 Pin Function Selection	
		PA6_MFP	Function
		0000	GPIOA[6]
		0010	ADC analog input6
		0011	Comparator0 output
		0100	SmartCard0 RST pin
		1001	Comparator0 charge/discharge path
[23:20]	PA5_MFP	PA.5 Pin Function Selection	
		PA5_MFP	Function
		0000	GPIOA[5]
		0010	ADC analog input5
		0011	Comparator0 N-end input0
		0100	SmartCard0 Power pin
		0101	I^{2}C1 data I/O pin
		0110	SPI1 1st slave select pin
		1001	Comparator0 charge/discharge path

Bits	Description		
[19:16]	**PA4_MFP**	**PA.4 Pin Function Selection**	
		PA4_MFP	**Function**
		0000	GPIOA[4]
		0010	ADC analog input4
		0011	Comparator0 P-end input0
		0100	SmartCard0 card detect pin
		1001	Comparator0 charge/discharge path
[15:12]	**PA3_MFP**	**PA.3 Pin Function Selection**	
		PA3_MFP	**Function**
		0000	GPIOA[3]
		0001	External interrupt 1
		0010	ADC analog input3
		0011	Comparator0 P-end input1
		0100	SmartCard0 DATA pin(SC0_UART_RXD)
		1001	Comparator0 charge/discharge path
[11:8]	**PA2_MFP**	**PA.2 Pin Function Selection**	
		PA2_MFP	**Function**
		0000	GPIOA[2]
		0001	External interrupt0 input pin
		0010	ADC analog input2
		0011	Comparator0 P-end input2
		0100	SmartCard0 clock pin(SC0_UART_TXD)
		1001	Comparator0 charge/discharge path
[7:4]	**PA1_MFP**	**PA.1 Pin Function Selection**	
		PA2_MFP	**Function**
		0000	GPIOA[1]
		0010	ADC analog input1
		0011	Comparator0 P-end input3
		1001	Comparator0 charge/discharge path
[3:0]	**PA0_MFP**	**PA.0 Pin Function Selection**	
		PA0_MFP	**Function**
		0000	GPIOA[0]
		0010	ADC input channel 0

Multiple Function Port A high byte Control Register (PA_H_MFP)

Register	Offset	R/W	Description	Reset Value
PA_H_MFP	GCR_BA+0x34	R/W	Port A High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PA15_MFP				PA14_MFP			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PA13_MFP				PA12_MFP			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PA11_MFP				PA10_MFP			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PA9_MFP				PA8_MFP			

Bits	Description		
[31:28]	**PA15_MFP**	**PA.15 Pin Function Selection**	
		PA15_MFP	**Function**
		0000	GPIOA[15]
		0010	Timer3 capture input
		0011	Comparator1 output
		0101	I^{2}C1 data I/O pin
		0110	SPI1 1st slave select pin
		1000	LCD segment output 16 at 48-pin LCD segment output 30 at 64-pin
[27:24]	**PA14_MFP**	**PA.14 Pin Function Selection**	
		PA14_MFP	**Function**
		0000	GPIOA[14]
		0101	I^{2}C1 clock pin
		0110	SPI1 serial clock pin
		1000	LCD segment output 17 at 48-pin LCD segment output 31 at 64-pin
		1001	Comparator0 charge/discharge path
[23:20]	**PA13_MFP**	**PA.13 Pin Function Selection**	
		PA13_MFP	**Function**
		0000	GPIOA[13]
		0011	Comparator1 N-end input

Bits	Description		
		0101	I^{2}C0 data I/O pin
		0110	SPI1 1st MISO (Master In, Slave Out) pin
		0111	UART0 Data receiver input pin
		1000	LCD segment output 18 at 48-pin
[19:16]	**PA12_MFP**	**PA.12 Pin Function Selection**	
		PA12_MFP	**Function**
		0000	GPIOA[12]
		0011	Comparator1 P-end input
		0101	I^{2}C 0 clock pin
		0110	SPI1 1st MOSI (Master Out, Slave In) pin
		0111	UART0 Data transmitter output pin(This pin could modulate with PWM0 output. Please refer PWM_SEL(UARTx_CTL[26:24]))
		1000	LCD segment output 19 at 48-pin LCD segment output 18 at 48-pin
[15:12]	**PA11_MFP**	**PA.11 Pin Function Selection**	
		PA13_MFP	**Function**
		0000	GPIOA[11]
		0010	ADC external trigger input.
		0100	SmartCard0 DATA pin(SC0_UART_RXD)
[11:8]	**PA10_MFP**	**PA.10 Pin Function Selection**	
		PA10_MFP	**Function**
		0100	SmartCard0 clock pin(SC0_UART_TXD)
		0000	GPIOA[10]
[7:4]	**PA9_MFP**	**PA.9 Pin Function Selection**	
		PA9_MFP	**Function**
		0000	GPIOA[9]
		0100	SmartCard0 RST pin
[3:0]	**PA8_MFP**	**PA.8 Pin Function Selection**	
		PA8_MFP	**Function**
		0000	GPIOA[8]
		0100	SmartCard0 Power pin

Multiple Function Port B low byte Control Register (PB_L_MFP)

Register	Offset	R/W	Description	Reset Value
PB_L_MFP	GCR_BA+0x38	R/W	Port B Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PB7_MFP				PB6_MFP			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PB5_MFP				PB4_MFP			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PB3_MFP				PB2_MFP			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PB1_MFP				PB0_MFP			

Bits	Description		
[31:28]	**PB7_MFP**	**PB.7 Pin Function Selection**	
		PB7_MFP	**Function**
		0000	GPIOB[7]
		0100	SmartCard0 card detect
		0111	UART1 Clear to Send input pin
		1000	LCD segment output 33 at 100-pin
[27:24]	**PB6_MFP**	**PB.6 Pin Function Selection**	
		PB6_MFP	**Function**
		0000	GPIOB[6]
		0001	Frequency Divider0 output pin
		0110	SPI1 2nd slave select pin
		0111	UART1 Data transmitter output pin(This pin could modulate with PWM0 output. Please refer PWM_SEL(UARTx_CTL[26:24]))
		1000	LCD segment output 25 at 64-pin LCD segment output 34 at 100-pin
[23:20]	**PB5_MFP**	**PB.5 Pin Function Selection**	
		PB5_MFP	**Function**
		0000	GPIOB[5]
		0110	SPI1 2nd MOSI (Master Out, Slave In) pin SmartCard0 RST
		0111	UART1 Data receiver input pin

Bits	Description		
		1000	LCD segment output 35 at 100-pin
[19:16]	**PB4_MFP**	**PB.4 Pin Function Selection**	
		PB4_MFP	**Function**
		0000	GPIOB[4]
		0110	SPI1 2nd MISO (Master In, Slave Out) pin
		0111	UART1 Request to Send output pin
[15:12]	**PB3_MFP**	**PB.3 Pin Function Selection**	
		PB3_MFP	**Function**
		0000	GPIOB[3]
		0010	Timer2 external counter input
		0101	I^{2}C0 data I/O pin
		0110	SPI1 2nd MISO (Master In, Slave Out) pin
		0111	UART0 Clear to Send input pin
		1000	LCD segment output 26 at 64-pin
[11:8]	**PB2_MFP**	**PB.2 Pin Function Selection**	
		PB2_MFP	**Function**
		0000	GPIOB[2]
		0010	Timer3 external counter input
		0101	I^{2}C0 clock pin
		0110	SPI1 2nd MOSI (Master Out, Slave In) pin
		0111	UART0 Request to Send output pin
		1000	LCD segment output 27 at 64-pin
[7:4]	**PB1_MFP**	**PB.1 Pin Function Selection**	
		PB1_MFP	**Function**
		0000	GPIOB[1]
		0001	External interrupt1 input pin
		0010	Timer 2 capture input
		0111	UART0 Data receiver input pin
		1000	LCD segment output 28 at 64-pin
[3:0]	**PB0_MFP**	**PB.0 Pin Function Selection**	
		PB0_MFP	**Function**
		0000	GPIOB[0]
		0001	Frequency Divider1 output pin
		0111	UART0 Data transmitter output pin(This pin could modulate with PWM0 output. Please refer PWM_SEL(UARTx_CTL[26:24]))

Bits	Description		
		1000	LCD segment output 29 at 64-pin

Multiple Function Port B high byte Control Register (PB_H_MFP)

Register	Offset	R/W	Description	Reset Value
PB_H_MFP	GCR_BA+0x3C	R/W	Port B High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PB15_MFP				PB14_MFP			
23	22	21	20	19	18	17	16
PB13_MFP				PB12_MFP			
15	14	13	12	11	10	9	8
PB11_MFP				PB10_MFP			
7	6	5	4	3	2	1	0
PB9_MFP				PB8_MFP			

Bits	Description		
[31:28]	PB15_MFP	PB.15 Pin Function Selection	
		PB15_MFP	Function
		0000	GPIOB[15]
		0110	SPI0 1st slave select pin
		0111	UART0 Clear to Send input pin
		1000	LCD segment output 12 at 48-pin LCD segment output 19 at 64-pin LCD segment output 23 at 100-pin
[27:24]	PB14_MFP	PB.14 Pin Function Selection	
		PB14_MFP	Function
		0000	GPIOB[14]
		0110	SPI0 serial clock pin
		0111	UART0 Data transmitter output pin(This pin could modulate with PWM0 output)
		1000	LCD segment output 13 at 48-pin LCD segment output 20 at 64-pin LCD segment output 24 at 100-pin
[23:20]	PB13_MFP	PB.13 Pin Function Selection	
		PB13_MFP	Function
		0000	GPIOB[13]
		0110	SPI0 1st MISO (Master In, Slave Out) pin
		0111	UART0 Data receiver input pin

Bits	Description		
		1000	LCD segment output 14 at 48-pin LCD segment output 21 at 64-pin LCD segment output 25 at 100-pin
[19:16]	**PB12_MFP**	**PB.12 Pin Function Selection**	
		PB12_MFP	**Function**
		0000	GPIOB[12]
		0001	Frequency Divider0 output pin
		0010	Timer0 external counter input or Timer0 toggle out.
		0110	SPI0 1st MOSI (Master Out, Slave In) pin
		0111	UART0 Request to Send output pin
		1000	LCD segment output 15 at 48-pin LCD segment output 22 at 64-pin LCD segment output 26 at 100-pin
[15:12]	**PB11_MFP**	**PB.11 Pin Function Selection**	
		PB11_MFP	**Function**
		0000	GPIOB[11]
		0010	Timer1 external counter input or Timer1 toggle out
		0110	SPI0 2rd MISO (Master In, Slave Out) pin
		0111	UART1 Request to Send output pin
		1000	LCD segment output 23 at 64-pin LCD segment output 27 at 100-pin
[11:8]	**PB10_MFP**	**PB.10 Pin Function Selection**	
		PB10_MFP	**Function**
		0000	GPIOB[10]
		0110	SPI0 1st MOSI (Master Out, Slave In) pin
		0111	UART1 Data receiver input pin
		1000	LCD segment output 24 at 64-pin LCD segment output 28 at 100-pin
[7:4]	**PB9_MFP**	**PB.9 Pin Function Selection**	
		PB9_MFP	**Function**
		0000	GPIOB[9]
		0011	PWM0 Channel1 output
		1000	LCD segment output 31 at 100-pin
[3:0]	**PB8_MFP**	**PB.8 Pin Function Selection**	
		PB8_MFP	**Function**
		0000	GPIOB[8]
		0001	External interrupt1 input pin

Bits	Description		
		0010	Timer0 external counter input or Timer0 toggle out.
		0011	PWM0 Channel0 output
		0100	Snooper pin
		1000	LCD segment output 32 at 100-pin

Multiple Function Port C low byte Control Register (PC_MFP)

Register	Offset	R/W	Description	Reset Value
PC_L_MFP	GCR_BA+0x40	R/W	Port C Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PC7_MFP				PC6_MFP			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PC5_MFP				PC4_MFP			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PC3_MFP				PC2_MFP			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PC1_MFP				PC0_MFP			

Bits	Description		
[31:28]	**PC7_MFP**	**PC.7 Pin Function Selection**	
		PC7_MFP	**Function**
		0000	GPIOC[7]
		0100	SmartCard0 Power pin
		0111	UART1 Data receiver input pin
		1000	LCD segment output 4 at 48-pin LCD segment output 11 at 64-pin LCD segment output 15 at 100-pin
[27:24]	**PC6_MFP**	**PC.6 Pin Fuction Selection**	
		PC6_MFP	**Function**
		0000	GPIOC[6]
		0100	SmartCard0 DATA pin(SC0_UART_RXD)
		0111	UART1 Request to Send output pin
		1000	LCD segment output 5 at 48-pin LCD segment output 12 at 64-pin LCD segment output 16 at 100-pin
[23:20]	**PC5_MFP**	**PC.5 Pin Function Selection**	
		PC5_MFP	**Function**
		0000	GPIOC[5]
		0100	SmartCard0 card detect pin
		1000	LCD segment output 6 at 48-pin LCD segment output 13 at 64-pin

NANO102/112 SERIES TECHINICAL REFERENCE MANUAL

Bits	Description		
			LCD segment output 17 at 100-pin
[19:16]	**PC4_MFP**	**PC.4 Pin Function Selection**	
		PC4_MFP	**Function**
		0000	GPIOC[4]
		0001	External interrupt0 input pin
		0100	SmartCard0 clock pin(SC0_UART_TXD)
		0111	UART1 Clear to Send input pin
		1000	LCD segment output 7 at 48-pin LCD segment output 14 at 64-pin LCD segment output 18 at 100-pin
[15:12]	**PC3_MFP**	**PC.3 Pin Function Selection**	
		PC3_MFP	**Function**
		0000	GPIOC[3]
		0011	PWM0 Channel3 output
		0101	I^{2}C1 data I/O pin
		1000	LCD segment output 8 at 48-pin LCD segment output 15 at 64-pin LCD segment output 19 at 100-pin
[11:8]	**PC2_MFP**	**PC.2 Pin Function Selection**	
		PC2_MFP	**Function**
		0000	GPIOC[2]
		0011	PWM0 Channel2 output
		0101	I^{2}C1 clock pin
		1000	LCD segment output 9 at 48-pin LCD segment output 16 at 64-pin LCD segment output 20 at 100-pin
[7:4]	**PC1_MFP**	**PC.1 Pin Function Selection**	
		PC1_MFP	**Function**
		0000	GPIOC[1]
		0011	PWM0 Channel1 output
		0101	I^{2}C0 data I/O pin
		1000	LCD segment output 10 at 48-pin LCD segment output 17 at 64-pin LCD segment output 21 at 100-pin
[3:0]	**PC0_MFP**	**PC.0 Pin Function Selection**	
		PC0_MFP	**Function**
		0000	GPIOC[0]

Bits	Description		
		0011	PWM0 Channel0 output
		0101	I^{2}C0 clock pin
		0110	SPI0 2nd slave select pin
		1000	LCD segment output 11 at 48-pin LCD segment output 18 at 64-pin LCD segment output 24 at 100-pin

Multiple Function Port C high byte Control Register (PC_H_MFP)

Register	Offset	R/W	Description	Reset Value
PC_H_MFP	GCR_BA+0x44	R/W	Port C High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PC15_MFP				PC14_MFP			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PC13_MFP				PC12_MFP			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PC11_MFP				PC10_MFP			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PC9_MFP				PC8_MFP			

Bits	Description		
[31:28]	**PC15_MFP**	**PC.15 Pin Function Selection**	
		PC15_MFP	**Function**
		0000	GPIOC[15]
		0100	SmartCard1 PWR pin
		1000	LCD segment output 0 at 48-pin LCD segment output 7 at 64-pin LCD segment output 7 at 100-pin
[27:24]	**PC14_MFP**	**PC.14 Pin Function Selection**	
		PC14_MFP	**Function**
		0000	GPIOC[14]
		0100	SmartCard1 card detect
		1000	LCD segment output 1 at 48-pin LCD segment output 8 at 64-pin LCD segment output 8 at 100-pin
[23:20]	**PC13_MFP**	**PC.13 Pin Function Selection**	
		PC13_MFP	**Function**
		0000	GPIOC[13]
		0100	SmartCard1 DATA pin(SC1_UART_RXD)
		1000	LCD segment output 9 at 100-pin
[19:16]	**PC12_MFP**	**PC.12 Pin Function Selection**	
		PC12_MFP	**Function**
		0000	GPIOC[12]

Bits	Description		
		0100	SmartCard1 clock pin(SC1_UART_TXD)
		1000	LCD segment output 10 at 100-pin
[15:12]	**PC11_MFP**	**PC.11 Pin Function Selection**	
		PC11_MFP	**Function**
		0000	GPIOC[11]
		0100	SmartCard1 PWR pin
		0101	I^{2}C 1 data I/O pin
		1000	LCD segment output 11 at 100-pin
[11:8]	**PC10_MFP**	**PC.10 Pin Function Selection**	
		PC10_MFP	**Function**
		0000	GPIOC[10]
		0100	SmartCard1 card detect
		0101	I^{2}C1 clock pin
		1000	LCD segment output 12 at 100-pin
[7:4]	**PC9_MFP**	**PC.9 Pin Function Selection**	
		PC9_MFP	**Function**
		1000	LCD segment output 2 at 48-pin LCD segment output 9 at 64-pin LCD segment output 13 at 100-pin
		0000	GPIOC[9]
[3:0]	**PC8_MFP**	**PC.8 Pin Function Selection**	
		PC8_MFP	**Function**
		0000	GPIOC[8]
		0100	SmartCard0 RST pin
		0111	UART1 Data transmitter output pin(This pin could modulate with PWM0 output. Please refer PWM_SEL(UARTx_CTL[26:24]))
		1000	LCD segment output 3 at 48-pin LCD segment output 10 at 64-pin LCD segment output 14 at 100-pin

Multiple Function Port D low byte Control Register (PD_L_MFP)

Register	Offset	R/W	Description	Reset Value
PD_L_MFP	GCR_BA+0x48	R/W	Port D Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PD7_MFP				PD6_MFP			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PD5_MFP				PD4_MFP			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PD3_MFP				PD2_MFP			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PD1_MFP				PD0_MFP			

Bits	Description		
[31:28]	**PD7_MFP**	**PD.7 Pin Function Selection**	
		PD7_MDP	**Function**
		0000	GPIOD[7]
		0100	SmartCard1 clock pin(SC1_UART_TXD)
		1000	LCD common output 3 at 48-pin LCD common output 3 at 64-pin LCD common output 3 at 100-pin
[27:24]	**PD6_MFP**	**PD.6 Pin Function Selection**	
		PD6_MFP	**Function**
		0000	GPIOD[6]
		1000	LCD segment output 0 at 64-pin(or as LD_COM4) LCD segment output 0 at 100-pin(or as LD_COM4)
[23:20]	**PD5_MFP**	**PD.5 Pin Function Selection**	
		PD5_MFP	**Function**
		0000	GPIOD[5]
		1000	LCD segment output 1 at 64-pin(or as LD_COM5) LCD segment output 1 at 100-pin(or as LD_COM5)
[19:16]	**PD4_MFP**	**PD.4 Pin Function Selection**	
		PD4_MFP	**Function**
		0000	GPIOD[4]
		0100	SmartCard1 RST pin
		1000	LCD segment output 2 at 64-pin

Bits	Description		
			LCD segment output 2 at 100-pin
[15:12]	**PD3_MFP**	**PD.3 Pin Function Selection**	
		PD3_MFP	**Function**
		0000	GPIOD[3]
		1000	LCD segment output 3 at 64-pin LCD segment output 3 at 100-pin
[11:8]	**PD2_MFP**	**PD.2 Pin Function Selection**	
		PD2_MFP	**Function**
		0000	GPIOD[2]
		1000	LCD segment output 4 at 64-pin LCD segment output 4 at 100-pin
[7:4]	**PD1_MFP**	**PD.1 Pin Function Selection**	
		PD1_MFP	**Function**
		0000	GPIOD[1]
		1000	LCD segment output 5 at 64-pin LCD segment output 5 at 100-pin
[3:0]	**PD0_MFP**	**PD.0 Pin Function Selection**	
		PD0_MFP	**Function**
		0000	GPIOD[0]
		1000	LCD segment output 6 at 64-pin LCD segment output 6 at 100-pin

Multiple Function Port D high byte Control Register (PD_H_MFP)

Register	Offset	R/W	Description	Reset Value
PD_H_MFP	GCR_BA+0x4C	R/W	Port D High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PD15_MFP				**PD14_MFP**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PD13_MFP				**PD12_MFP**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PD11_MFP				**PD10_MFP**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PD9_MFP				**PD8_MFP**			

Bits	Description		
[30:28]	**PD15_MFP**	**PD.15 Pin Function Selection**	
		PD15_MFP	**Function**
		0000	GPIOD[15]
		1000	LCD Unit voltage for LCD charge pump circuit at 48-pin LCD Unit voltage for LCD charge pump circuit at 64-pin LCD Unit voltage for LCD charge pump circuit at 100-pin
[27:24]	**PD14_MFP**	**PD.14 Pin Function Selection**	
		PD14_MFP	**Function**
		0000	GPIOD[14]
		1000	LCD Unit voltage for LCD charge pump circuit at 48-pin LCD Unit voltage for LCD charge pump circuit at 64-pin LCD Unit voltage for LCD charge pump circuit at 100-pin
[23:20]	**PD13_MFP**	**PD.13 Pin Function Selection**	
		PD13_MFP	**Function**
		0000	GPIOD[13]
		0001	External interrupt 1 input pin
		1000	LCD Unit voltage for LCD charge pump circuit at 48-pin LCD Unit voltage for LCD charge pump circuit at 64-pin

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td><table><tr><td></td><td>LCD Unit voltage for LCD charge pump circuit at 100-pin</td></tr></table></td></tr>
<tr><td>[19:16]</td><td><b>PD12_MFP</b></td><td><b>PD.12 Pin Function Selection</b>
<table>
<tr><th>PD12_MFP</th><th>Function</th></tr>
<tr><td>0000</td><td>GPIOD[12]</td></tr>
<tr><td>0001</td><td>Frequency Divider0 output pin</td></tr>
<tr><td>0010</td><td>Timer1 external counter input or Timer1 toggle out</td></tr>
<tr><td>0011</td><td>PWM0 Channel0 output</td></tr>
<tr><td>1000</td><td>LCD externl capacitor pin of charge pump circuit at 64-pin
LCD externl capacitor pin of charge pump circuit at 100-pin</td></tr>
<tr><td>1001</td><td>1, 1/2, 1/4, 1/16 Hz clock output</td></tr>
</table></td></tr>
<tr><td>[15:12]</td><td><b>PD11_MFP</b></td><td><b>PD.11 Pin Function Selection</b>
<table>
<tr><th>PD11_MFP</th><th>Function</th></tr>
<tr><td>0000</td><td>GPIOD[11]</td></tr>
<tr><td>0010</td><td>Timer0 capture input</td></tr>
<tr><td>0011</td><td>PWM0 Channel1 output</td></tr>
<tr><td>1000</td><td>LCD externl capacitor pin of charge pump circuit at 64-pin
LCD externl capacitor pin of charge pump circuit at 100-pin</td></tr>
</table></td></tr>
<tr><td>[11:8]</td><td><b>PD10_MFP</b></td><td><b>PD.10 Pin Function Selection</b>
<table>
<tr><th>PD10_MFP</th><th>Function</th></tr>
<tr><td>0000</td><td>GPIOD[10]</td></tr>
<tr><td>0010</td><td>Timer1 capture input</td></tr>
<tr><td>0011</td><td>PWM0 Channel2 output</td></tr>
<tr><td>1000</td><td>LCD common output 0 at 48-pin
LCD common output 0 at 64-pin
LCD common output 0 at 100-pin</td></tr>
</table></td></tr>
<tr><td>[7:4]</td><td><b>PD9_MFP</b></td><td><b>PD.9 Pin Function Selection</b>
<table>
<tr><th>PD9_MFP</th><th>Function</th></tr>
<tr><td>0000</td><td>GPIOD[9]</td></tr>
<tr><td>0011</td><td>PWM0 Channel3 output</td></tr>
<tr><td>0100</td><td>SmartCard1 RST pin</td></tr>
<tr><td>1000</td><td>LCD common output 1 at 48-pin
LCD common output 1 at 64-pin
LCD common output 1 at 100-pin</td></tr>
</table></td></tr>
<tr><td>[3:0]</td><td><b>PD8_MFP</b></td><td><b>PD.8 Pin Function Selection</b></td></tr>
</table>

Bits	Description		
		PD8_MFP	**Function**
		0000	GPIOD[8]
		0100	SmartCard1 DATA pin(SC1_UART_RXD)
		1000	LCD common output 2 at 48-pin LCD common output 2 at 64-pin LCD common output 2 at 100-pin

Multiple Function Port E low byte Control Register (PE_L_MFP)

Register	Offset	R/W	Description	Reset Value
PE_L_MFP	GCR_BA+0x50	R/W	Port E Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PE7_MFP				PE6_MFP			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PE5_MFP				PE4_MFP			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PE3_MFP				PE2_MFP			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PE1_MFP				PE0_MFP			

Bits	Description		
[31:28]	**PE7_MFP**	**PE.7 Pin Function Selection**	
		PE7_MFP	**Function**
		0000	GPIOE[7]
		0100	SmartCard1 DATA pin(SC1_UART_RXD)
[27:24]	**PE6_MFP**	**PE.6 Pin Function Selection**	
		PE6_MFP	**Function**
		0000	GPIOE[6]
		0100	SmartCard1 clock pin(SC1_UART_TXD)
[23:20]	**PE5_MFP**	**PE.5 Pin Function Selection**	
		PE5_MFP	**Function**
		0000	GPIOE[5]
		0100	SmartCard1 PWR pin
[19:16]	**PE4_MFP**	**PE.4 Pin Function Selection**	
		PE4_MFP	**Function**
		0000	GPIOE[4]
		0100	SmartCard1 RST pin
[15:12]	**PE3_MFP**	**PE.3 Pin Function Selection**	
		PE3_MFP	**Function**
		0000	GPIOE[3]
		0110	SPI0 1st slave select pin
[11:8]	**PE2_MFP**	**PE.2 Pin Function Selection**	

Bits	Description		
		PE2_MFP	**Function**
		0000	GPIOE[2]
		0110	SPI0 serial clock pin
[7:4]	**PE1_MFP**	**PE.1 Pin Function Selection**	
		PE1_MFP	**Function**
		0000	GPIOE[1]
		0110	SPI0 1st MISO (Master In, Slave Out) pin
[3:0]	**PE0_MFP**	**PE.0 Pin Function Selection**	
		PE0_MFP	**Function**
		0000	GPIOE[0]
		0110	SPI0 1st MOSI (Master Out, Slave In) pin

Multiple Function Port E high byte Control Register (PE_H_MFP)

Register	Offset	R/W	Description	Reset Value
PE_H_MFP	GCR_BA+0x54	R/W	Port E High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				**Reserved**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				**Reserved**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**Reserved**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PE9_MFP				**PE8_MFP**			

Bits	Description		
[31:8]	**Reserved**	Reserved.	
[7:4]	**PE9_MFP**	**PE.9 Pin Function Selection**	
		PE9_MFP	**Function**
		0000	GPIOE[9]
		0011	PWM0 Channel3 output
		1000	LCD segment output 29 at 100-pin
[3:0]	**PE8_MFP**	**PE.8 Pin Function Selection**	
		PE8_MFP	**Function**
		0000	GPIOE[8]
		0011	PWM0 Channel2 output
		1000	LCD segment output 30 at 100-pin

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

Multiple Function Port F low byte Control Register (PF_L_MFP)

Register	Offset	R/W	Description	Reset Value
PF_L_MFP	GCR_BA+0x58	R/W	Port F Low Byte Multiple Function Control Register	0x00FF_FFFF

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PF5_MFP				**PF4_MFP**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PF3_MFP				**PF2_MFP**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PF1_MFP				**PF0_MFP**			

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:24]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
<tr><td>[23:20]</td><td><b>PF5_MFP</b></td><td><b>PF.5 Pin Function Selection</b>
<table>
<tr><th>PF5_MFP</th><th>Function</th></tr>
<tr><td>0000</td><td>GPIOF[5]</td></tr>
<tr><td>0010</td><td>Timer0 capture input</td></tr>
<tr><td>0011</td><td>PWM0 Channel3 output</td></tr>
<tr><td>1001</td><td>Comparator0 charge/discharge path</td></tr>
<tr><td>1111</td><td>Serial Wired Debugger Data pin</td></tr>
</table></td></tr>
<tr><td>[19:16]</td><td><b>PF4_MFP</b></td><td><b>PF.4 Pin Function Selection</b>
<table>
<tr><th>PF4_MFP</th><th>Function</th></tr>
<tr><td>0000</td><td>GPIOF[4]</td></tr>
<tr><td>0001</td><td>Frequency Divider1 output pin</td></tr>
<tr><td>0010</td><td>Timer1 capture input</td></tr>
<tr><td>0011</td><td>PWM0 Channel2 output</td></tr>
<tr><td>1001</td><td>1, 1/2, 1/4, 1/8, 1/16 Hz clock output</td></tr>
<tr><td>1111</td><td>Serial Wired Debugger Clock pin</td></tr>
</table></td></tr>
<tr><td>[15:12]</td><td><b>PF3_MFP</b></td><td><b>PF.3 Pin Function Selection</b>
<table>
<tr><th>PF3_MFP</th><th>Function</th></tr>
<tr><td>0000</td><td>GPIOF[3]</td></tr>
<tr><td>0001</td><td>External interrupt0 input pin</td></tr>
</table></td></tr>
</table>

Bits	Description		
		0010	Timer 2 capture input
		0111	UART1 Data transmitter output pin(This pin could modulate with PWM0 output. Please refer PWM_SEL(UARTx_CTL[26:24]))
		1111	External 4~24 MHz crystal output pin
[11:8]	**PF2_MFP**	**PF.2 Pin Function Selection**	
		PF2_MFP	**Function**
		0000	GPIOF[2]
		0001	External interrupt1 input pin
		0010	Timer3 capture input
		0111	UART1 Data receiver input pin
		1111	External 4~24 MHz crystal input pin(default)
[7:4]	**PF1_MFP**	**PF.1 Pin Function Selection**	
		PF1_MFP	**Function**
		0000	GPIOF[1]
		0010	Timer2 external counter input or Timer2 toggle out.
		1111	External 32.768 kHz crystal output pin(default)
[3:0]	**PF0_MFP**	**PF.0 Pin Function Selection**	
		PF0_MFP	**Function**
		0000	GPIOF[1]
		0010	Timer3 external counter input or Timer3 toggle out.
		1111	External 32.768 kHz crystal input pin(default)

Power-On-Reset Control Register (PORCTL)

Register	Offset	R/W	Description	Reset Value
PORCTL	GCR_BA+0x60	R/W	Power-On-Reset Controller Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
POR_DIS_CODE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
POR_DIS_CODE							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**POR_DIS_CODE**	**Power-on Reset Enable Control** This is a protected register. Please refer to open lock sequence to program it. When powered on, the POR circuit generates a reset signal to reset the whole chip function, but noise on the power may cause the POR active again. If setting the POR_DIS_CODE to 0x5AA5, the POR reset function will be disabled and the POR function will be active again when POR_DIS_CODE is set to another value or POR_DIS_CODE is reset by chip other reset functions, including: /RESET, Watchdog Timer reset, BOD reset, ICE reset command and the software-chip reset function

Brown-out Detect Control Register (BODCTL)

A part of the BODCTL control registers bits are initiated by the flash configuration

Register	Offset	R/W	Description	Reset Value
BODCTL	GCR_BA+0x64	R/W	Brown-out Detector Controller Register	0x00xx_x0xx

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
BOD25_TRIM				**BOD20_TRIM**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BOD17_TRIM				**Reserved**	**BOD25_INT_EN**	**BOD20_INT_EN**	**BOD17_INT_EN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**BOD25_RST_EN**	**BOD20_RST_EN**	**BOD17_RST_EN**	**Reserved**	**BOD25_EN**	**BOD20_EN**	**BOD17_EN**

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:20]	**BOD25_TRIM**	**BOD 2.5 TRIM Value** This is a protected register. Please refer to open lock sequence to program it. This value is used to control BOD25 detect voltage level, nominal 2.5 V. Higher trim value, higher detection voltage.
[19:16]	**BOD20_TRIM**	**BOD 2.0 TRIM Value** This is a protected register. Please refer to open lock sequence to program it. This value is used to control BOD20 detect voltage level, nominal 2.0 V. Higher trim value, higher detection voltage.
[15:12]	**BOD17_TRIM**	**BOD 1.7 TRIM Value** This is a protected register. Please refer to open lock sequence to program it. This value is used to control BOD17 detect voltage level, nominal 1.7 V. Higher trim value, higher detection voltage.
[11]	**Reserved**	Reserved.
[10]	**BOD25_INT_EN**	**BOD 2.5 V Interrupt Enable Control** This is a protected register. Please refer to open lock sequence to program it. 0 = Interrupt does not issue when BOD25 occurs. 1 = Interrupt issues when BOD25 occurs.
[9]	**BOD20_INT_EN**	**BOD 2.0 V Interrupt Enable Control** 0 = Interrupt does not issue when BOD20 occurs. 1 = Interrupt issues when BOD20 occurs.
[8]	**BOD17_INT_EN**	**BOD 1.7 V Interrupt Enable Control** This is a protected register. Please refer to open lock sequence to program it. 0 = Interrupt does not issue when BOD17 occurs. 1 = Interrupt issues when BOD17 occurs.

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[7]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
<tr><td>[6]</td><td><b>BOD25_RST_EN</b></td><td><b>BOD 2.5 V Reset Enable Control</b>
This is a protected register. Please refer to open lock sequence to program it.
0 = Reset does not issue when BOD25 occurs.
1 = Reset issues when BOD25 occurs.
The default value is set by flash controller user configuration register config0 bit[20:19]</td></tr>
<tr><td>[5]</td><td><b>BOD20_RST_EN</b></td><td><b>BOD 2.0 V Reset Enable Control</b>
This is a protected register. Please refer to open lock sequence to program it.
0 = Reset does not issue when BOD20 occurs.
1 = Reset issues when BOD20 occurs.
The default value is set by flash controller user configuration register config0 bit[20:19]</td></tr>
<tr><td>[4]</td><td><b>BOD17_RST_EN</b></td><td><b>BOD 1.7 V Reset Enable Control</b>
This is a protected register. Please refer to open lock sequence to program it.
0 = Reset does not issue when BOD17 occurs.
1 = Reset issues when BOD17 occurs.
The default value is set by flash controller user configuration register config0 bit[20:19]
BOD17_RST_EN can be controlled (enable or disable) only when BOD17_EN is high.
<table>
<tr><th></th><th>BOD17 RST Function Enable</th></tr>
<tr><td>BOD17_EN low</td><td>Enabled</td></tr>
<tr><td>BOD17_EN high</td><td>BOD17_RST_EN</td></tr>
</table></td></tr>
<tr><td>[3]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
<tr><td>[2]</td><td><b>BOD25_EN</b></td><td><b>Brown-out Detector 2.5 V Function Enable Control</b>
This is a protected register. Please refer to open lock sequence to program it.
0 = Brown-out Detector 2.5 V function Disabled.
1 = Brown-out Detector 2.5 V function Enabled.</td></tr>
<tr><td>[1]</td><td><b>BOD20_EN</b></td><td><b>Brown-out Detector 2.0 V Function Enable Control</b>
This is a protected register. Please refer to open lock sequence to program it.
0 = Brown-out Detector 2.0 V function Disabled.
1 = Brown-out Detector 2.0 V function Enabled.
BOD20_EN is default on. If SW disables it, Brown-out Detector 2.0 V function is not disabled until chip enters Power-down mode. If system is not in Power-down mode, BOD20_EN will be enabled by hardware automatically.</td></tr>
<tr><td>[0]</td><td><b>BOD17_EN</b></td><td><b>Brown-out Detector 1.7V Function Enable Control</b>
This is a protected register. Please refer to open lock sequence to program it.
The default value is set by flash controller user configuration register config0 bit[20:19]
Users can disable BOD17_EN but it takes effective (disabled) only in Power-down mode. Once existing Power-down mode, BOD17 will be enabled by HW automatically.
When CPU reads this bit, CPU will read whether BOD17 function enabled or not. In other words,CPU will always read high.
0 = Brown-out Detector 1.7V function Disabled.
1 = Brown-out Detector 1.7V function Enabled.
<table>
<tr><th></th><th>BOD17 Function Enable</th></tr>
<tr><td>Normal mode</td><td>Enabled</td></tr>
<tr><td>Power-down mode</td><td>BOD17_EN</td></tr>
</table></td></tr>
</table>

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

Brown-out Detector Status Register (BODSTS)

Register	Offset	R/W	Description	Reset Value
BODSTS	GCR_BA+0x68	R	Brown-out Detector Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**BOD25**	**BOD20**	**BOD17**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**BOD25_rise**	**BOD20_rise**	**BOD17_rise**	**BOD25_drop**	**BOD20_drop**	**BOD17_drop**	**BOD_INT**

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10]	**BOD25**	**Brown-out Detector 2.5V Status** This bit reflects the BOD25 status. BOD25 is high if detected voltage is higher than 2.5 V. BOD25 is low if detected voltage is lower than 2.5 V. **Note:** This bit is ready-only.
[9]	**BOD20**	**Brown-out Detector 2.0V Status** This bit reflects the BOD20 status. BOD20 is high if detected voltage is higher than 2.0 V. BOD20 is low if detected voltage is lower than 2.0 V. **Note:** This bit is ready-only.
[8]	**BOD17**	**Brown-out Detector 1.7V Status** This bit reflects the BOD17 status. BOD17 is high if detected voltage is higher than 1.7 V. BOD17 is low if detected voltage is lower than 1.7 V. **Note:** This bit is ready-only.
[7]	**Reserved**	Reserved.
[6]	**BOD25_rise**	**Brown-out Detector Higher Than 2.5V Status** Setting BOD25_rise high means once the detected voltage is higher than target detected voltage setting (2.5V). Software can write 1 to clear BOD25_rise.
[5]	**BOD20_rise**	**Brown-out Detector Higher Than 2.0V Status** Setting BOD20_rise high means once the detected voltage is higher than target detected voltage setting (2.0V). Software can write 1 to clear BOD20_rise
[4]	**BOD17_rise**	**Brown-out Detector Higher Than 1.7V Status** Setting BOD17_rise high means once the detected voltage is higher than target detected voltage setting (1.7V). Software can write 1 to clear BOD17_rise
[3]	**BOD25_drop**	**Brown-out Detector Lower Than 2.5V Status** Setting BOD25_drop high means once the detected voltage is lower than target detected voltage setting (2.5V). Software can write 1 to clear BOD25_drop
[2]	**BOD20_drop**	**Brown-out Detector Lower Than 2.0V Status**

		Setting BOD20_drop high means once the detected voltage is lower than target detected voltage setting (2.0V). Software can write 1 to clear BOD20_drop
[1]	**BOD17_drop**	**Brown-out Detector Lower Than 1.7V Status** Setting BOD17_drop high means once the detected voltage is lower than target detected voltage setting (1.7V). Software can write 1 to clear BOD17_drop
[0]	**BOD_INT**	**Brown-out Detector Interrupt Status** 0 = Brown-out Detector does not detect any voltage drift at V_{DD} down through or up through the target detected voltage after interrupt is enabled. 1 = When Brown-out Detector detects the V_{DD} is dropped down through the target detected voltage or the V_{DD} is raised up through the target detected voltage and Brown-out interrupt is enabled, this bit will be set to 1. This bit is cleared by writing 1 to it.

Internal Voltage Reference Control Register (Int_VREFCTL)

Register	Offset	R/W	Description	Reset Value
Int_VREFCTL	GCR_BA+0x6C	R/W	Internal Voltage Reference Control Register	0x0000_0700

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**VREF_TRIM**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**EXT_MODE**	**SEL25**		**REG_EN**	**BGP_EN**

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11:8]	**VREF_TRIM**	**Internal Voltage Reference Trim**
[7:5]	**Reserved**	Reserved.
[4]	**EXT_MODE**	**Regulator External Mode** This is a protected register. Please refer to open lock sequence to program it. Users can output regulator output voltage in V_{REF} pin if EXT_MODE is high. 0 = No connection with external V_{REF} pin. 1 = Connet to external V_{REF} pin. Connect a 1uF to 10uF capacitor to AV_{SS} will let internal voltage reference be more stable.
[3:2]	**SEL25**	**Regulator Output Voltage Selection** Select internal reference voltage level. This is a protected register. Please refer to open lock sequence to program it. 00 = 1.5V. 01 = 1.8V. 10 = 2.5V. 11 = 2.5V.
[1]	**REG_EN**	**Regulator Enable Control** Enable internal 1.5, 1.8V or 2.5V reference voltage. This is a protected register. Please refer to open lock sequence to program it. 0 = Disabled. 1 = Enabled.
[0]	**BGP_EN**	**Band-gap Enable Control** This is a protected register. Please refer to open lock sequence to program it. Band-gap is the reference voltage of internal reference voltage. User must enable band-gap if want to enable internal 1.5, 1.8V or 2.5V reference voltage. 0 = Disabled. 1 = Enabled.

LDO Control Register (LDO_CTL)

Register	Offset	R/W	Description	Reset Value
LDO_CTL	GCR_BA+0x70	R/W	LDO Control Register	0x0000_0X0C

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**LDO_LEVEL**		**Reserved**	**LDO_PD**

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3:2]	**LDO_LEVEL**	**LDO Output Voltage Select** This is a protected register. Please refer to open lock sequence to program it. 00 = Reseved. 01 = 1.6V. 10 = 1.8V. 11 = 1.8V.
[1]	**Reserved**	Reserved.
[0]	**LDO_PD**	**LDO Power Off** This is a protected register. Please refer to open lock sequence to program it. Set this bit high will off LDO and cause Chip in unexpected state. User must keep this bit low. 0 = LDO Enabled. 1 = LDO Disabled.

HIRC Trim Control Register (IRCTRIMCTL)

Register	Offset	R/W	Description	Reset Value
IRCTRIMCTL	GCR_BA+0x80	R/W	HIRC Trim Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							ERR_STOP
7	6	5	4	3	2	1	0
TRIM_RETRY_CNT		TRIM_LOOP		Reserved		TRIM_SEL	

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:8]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
<tr><td>[8]</td><td><b>ERR_STOP</b></td><td><b>Trim Stop When 32.768 KHz Error Detected</b>
This bit is used to control if stop the HIRC trim operation when 32.768 kHz clock error is detected.
If set this bit high and 32.768 kHz clock error detected, the status 32K_ERR_INT (IRCTRIMINT[2]) would be set high and HIRC trim operation was stopped. If this bit is low and 32.768 kHz clock error detected, the status 32K_ERR_INT (IRCTRIMINT[2]) would be set high and HIRC trim operation is continuously.
0 = Continue the HIRC trim operation even if 32.768 kHz clock error detected.
1 = Stop the HIRC trim operation if 32.768 kHz clock error detected.</td></tr>
<tr><td>[7:6]</td><td><b>TRIM_RETRY_CNT</b></td><td><b>Trim Value Update Limitation Count</b>
This field defines that how many times the auto trim circuit will try to update the HIRC trim value before the frequency of HIRC locked.
Once the HIRC locked, the internal trim value update counter will be reset.
If the trim value update counter reached this limitation value and frequency of HIRC still doesn’t lock, the auto trim operation will be disabled and TRIM_SEL (IRCTRIMCTL[1:0]) will be cleared to 00.
<table>
<tr><th>TRIM_RETRY_CNT</th><th>Trim Retry Count Limitation</th></tr>
<tr><td>00</td><td>Trim retry count limitation is 64</td></tr>
<tr><td>01</td><td>Trim retry count limitation is 128</td></tr>
<tr><td>10</td><td>Trim retry count limitation is 256</td></tr>
<tr><td>11</td><td>Trim retry count limitation is 512</td></tr>
</table></td></tr>
<tr><td>[5:4]</td><td><b>TRIM_LOOP</b></td><td><b>Trim Calculation Loop</b>
This field defines that trim value calculation is based on how many 32.768 kHz clock.
For example, if TRIM_LOOP is set as 00, auto trim circuit will calculate trim value based on the average frequency difference in 4 32.768 kHz clock.
<table>
<tr><th>TRIM_LOOP</th><th>Average Frequency Difference</th></tr>
</table></td></tr>
</table>

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td><table><tr><td>00</td><td>4 32.768 kHz clock</td></tr><tr><td>01</td><td>8 32.768 kHz clock</td></tr><tr><td>10</td><td>16 32.768 kHz clock</td></tr><tr><td>11</td><td>32 32.768 kHz clock</td></tr></table></td></tr>
<tr><td>[3:2]</td><td></td><td>Reserved.</td></tr>
<tr><td>[1:0]</td><td><b>TRIM_SEL</b></td><td><b>Trim Frequency Selection</b>
This field indicates the target frequency of HIRC auto trim.
If no any target frequency is selected (TRIM_SEL is 00), the HIRC auto trim function is disabled.
During auto trim operation, if 32.768 kHz clock error detected or trim retry limitation count reached, this field will be cleared to 00 automatically.<table><tr><th>TRIM_SEL</th><th>Function</th></tr><tr><td>00</td><td>Disable HIRC auto trim function</td></tr><tr><td>01</td><td>Enable HIRC auto trim function and trim HIRC to 11.0592 MHz</td></tr><tr><td>10</td><td>Enable HIRC auto trim function and trim HIRC to 12 MHz</td></tr><tr><td>11</td><td>Enable HIRC auto trim function and trim HIRC to 16 MHz</td></tr></table></td></tr>
</table>

HIRC Trim Interrupt Enable Register (IRCTRIMIEN)

Register	Offset	R/W	Description	Reset Value
IRCTRIMIEN	GCR_BA+0x84	R/W	HIRC Trim Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**32K_ERR_IEN**	**TRIM_FAIL_IEN**	**Reserved**

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**32K_ERR_IEN**	**32.768 KHz Clock Error Interrupt Enable Control** This bit controls if CPU would get an interrupt while 32.768 kHz clock is inaccuracy during auto trim operation. If this bit is high, and 32K_ERR_INT (IRCTRIMINT[2]) is set during auto trim operation, an interrupt will be triggered to notify the 32.768 kHz clock frequency is inaccuracy. 0 = 32K_ERR_INT (IRCTRIMINT[2]) status Disabled to trigger an interrupt to CPU. 1 = 32K_ERR_INT (IRCTRIMINT[2]) status Enabled to trigger an interrupt to CPU.
[1]	**TRIM_FAIL_IEN**	**Trim Failure Interrupt Enable Control** This bit controls if an interrupt will be triggered while HIRC trim value update limitation count reached and HIRC frequency still not locked on target frequency set by TRIM_SEL (IRCTRIMCTL[1:0]). If this bit is high and TRIM_FAIL_INT (IRCTRIMINT[1]) is set during auto trim operation, an interrupt will be triggered to notify that HIRC trim value update limitation count was reached. 0 = TRIM_FAIL_INT (IRCTRIMINT[1:0]) status Disabled to trigger an interrupt to CPU. 1 = TRIM_FAIL_INT (IRCTRIMINT[1:0]) status Enabled to trigger an interrupt to CPU.
[0]	**Reserved**	Reserved.

HIRC Trim Interrupt Status Register (IRCTRIMINT)

Register	Offset	R/W	Description	Reset Value
IRCTRIMINT	GCR_BA+0x88	R/W	HIRC Trim Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**32K_ERR_INT**	**TRIM_FAIL_INT**	**FREQ_LOCK**

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**32K_ERR_INT**	**32.768 KHz Clock Error Interrupt Status** This bit indicates that 32.768 kHz clock frequency is inaccuracy. Once this bit is set, the auto trim operation stopped and TRIM_SEL (IRCTRIMCTL[1:0]) will be cleared to 00 by hardware automatically. If this bit is set and 32K_ERR_IEN (IRCTRIMIEN[2]) is high, an interrupt will be triggered to notify the 32.768 kHz clock frequency is inaccuracy. Write 1 to clear this to zero. 0 = 32.768 kHz clock frequency is accuracy. 1 = 32.768 kHz clock frequency is inaccuracy.
[1]	**TRIM_FAIL_INT**	**Trim Failure Interrupt Status** This bit indicates that HIRC trim value update limitation count reached and HIRC clock frequency still doesn’t lock. Once this bit is set, the auto trim operation stopped and TRIM_SEL (IRCTRIMCTL[1:0]) will be cleared to 00 by hardware automatically. If this bit is set and TRIM_FAIL_IEN (IRCTRIMIEN[1]) is high, an interrupt will be triggered to notify that HIRC trim value update limitation count was reached. Write 1 to clear this to zero. 0 = Trim value update limitation count doesn’t reach. 1 = Trim value update limitation count reached and HIRC frequency still doesn’t lock.
[0]	**FREQ_LOCK**	**HIRC Frequency Lock Status** This bit indicates the HIRC frequency lock. This is a status bit and doesn’t trigger any interrupt.

Register Lock Key Address Register (RegLockAddr)

Some of the system control registers need to be protected to avoid inadvertent write and disturb the chip operation. These system control registers are locked after the power-on reset till users to open the lock. For users to program these protected registers, an open lock sequence needs to be followed by a special programming sequence. The open sequence is to continue write the data “59h”, “16h” “88h” to the key controller address 0x5000_0100. Any different data value or different sequence or any other write operations to any other address during these three data program aborts the whole sequence.

After the lock is opened, users can check the lock bit RegLockAddr[0]. “1” is unlocked, “0” is locked. Then users can update the target register value if RegUnLock is high and write any data to the address “0x5000_0100” to re-lock the protected registers

This register is written for open the RegUnLock key and read for the RegUnLock status.

Register	Offset	R/W	Description	Reset Value
RegLockAddr	GCR_BA+0x100	R/W	Register Lock Key address	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**RegUnLock**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**RegUnLock**	**Protected Register Enable Control** 0 = Protected register are Locked. Any write to the target register is ignored. 1 = Protected registers are Unlocked.

6.5 Clock Controller

6.5.1 Overview

The clock controller generates clocks for the whole chip, Including system clocks (CPU clock, HCLKx, and PCLKx) and all peripheral module clocks. HCLKx means AHB bus clock for peripherals on AHB bus. PCLKx means APB bus clock for peripherals on APB bus. PCLKx can be the same as HCLKx or divided from HCLKx. The clock controller also implements the power control function with the individually clock ON/OFF control, clock source selection and a 4-bit clock divider. The chip will not enter Power-down mode until CPU sets the power down enable bit PD_EN(PWRCTL[6]) and executes the WFI instruction. In the Power-down mode, clock controller turns off the external high frequency crystal, internal high frequency oscillator, and system clocks (CPU clock, HCLKx, and PCLKx) to reduce the power consumption.

The clock controller consists of 5 sources as listed below:

- 32.768 kHz external low speed crystal oscillator (LXT)
- 4~ 24 MHz external high speed crystal oscillator (HXT)
- 12/16 MHz internal high speed RC oscillator (HIRC)
- One programmable PLL FOUT (PLL source can be selected from HXT or HIRC)
- 10 kHz internal low speed RC oscillator (LIRC)

6.5.2 Features

- Generates clocks for system clocks and all peripheral module clocks.
- Each peripheral module clock can be turned on/off.
- High frequency crystal, internal high frequency oscillator, and system clocks will be turned off when chip is in Power-down mode.

6.5.3 Block Diagram

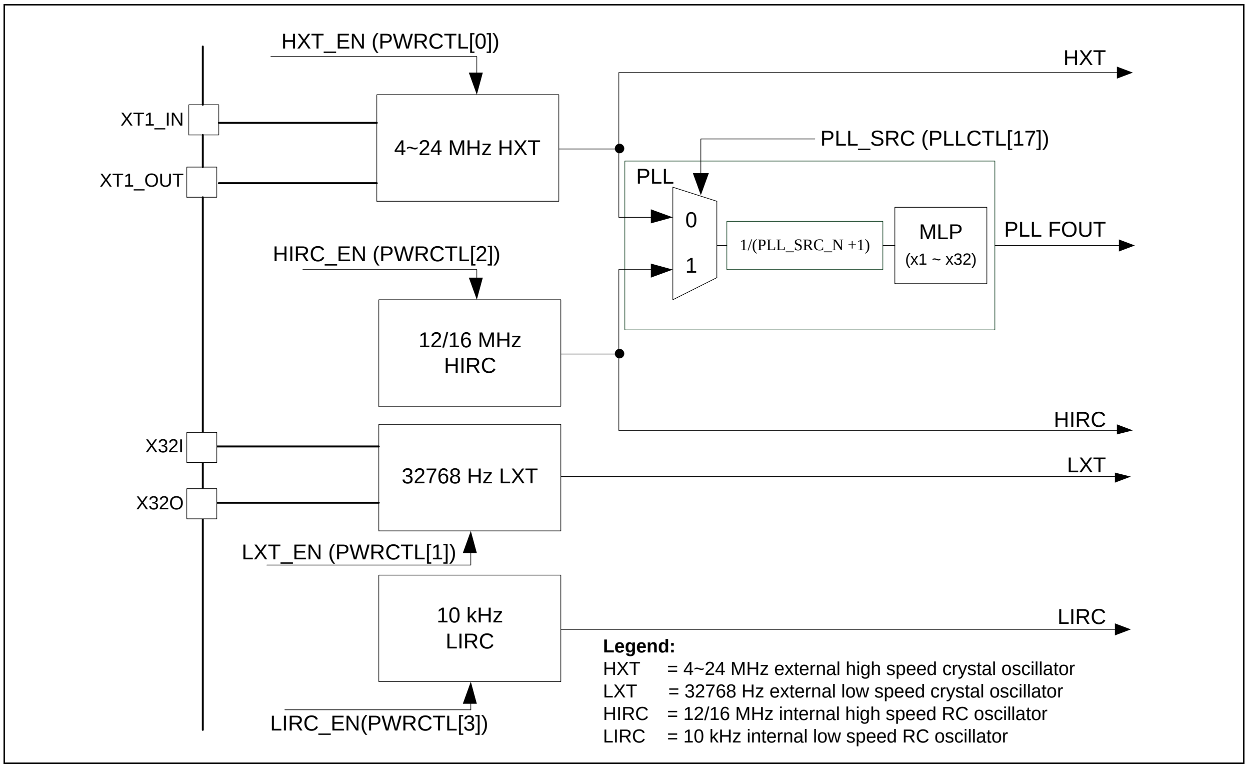
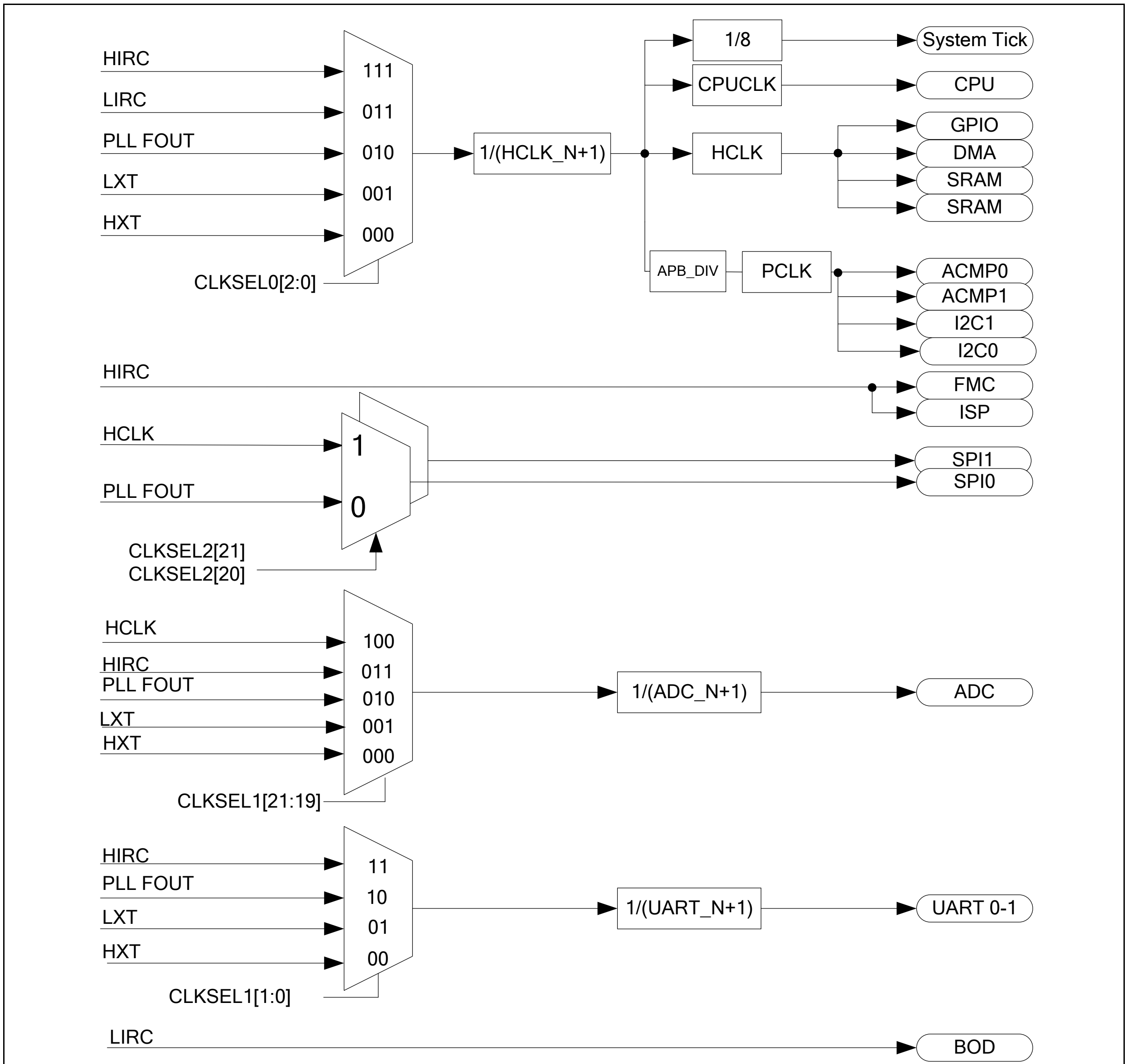


Figure 6-6 Clock Controller Block Diagram (1/3)

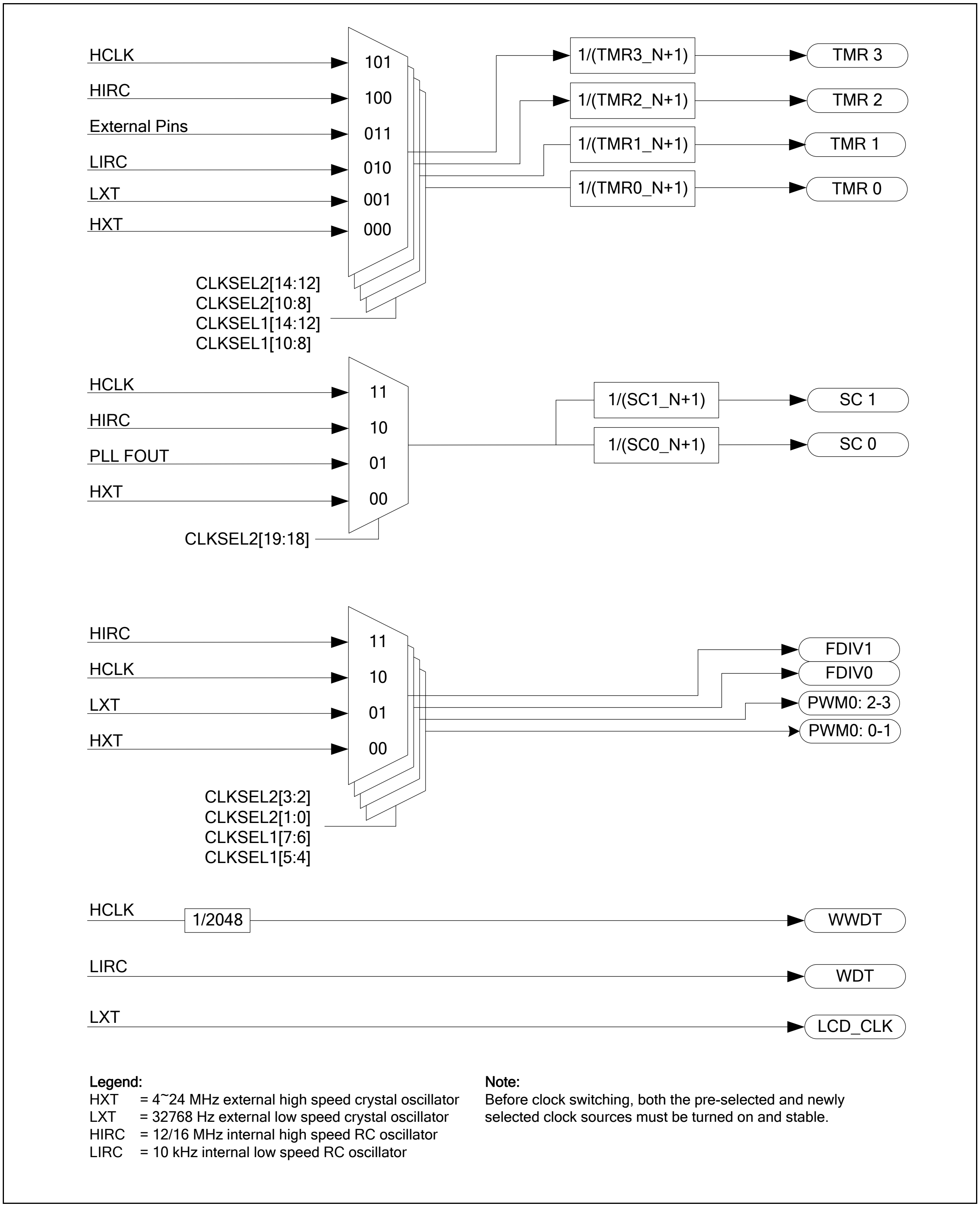


Legend:

HXT	= 4~24 MHz external high speed crystal oscillator
LXT	= 32768 Hz external low speed crystal oscillator
HIRC	= 12/16 MHz internal high speed RC oscillator
LIRC	= 10 kHz internal low speed RC oscillator

Note:
Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.

Figure 6-7 Clock Controller Block Diagram (2/3)



Legend:

HXT = 4~24 MHz external high speed crystal oscillator
LXT = 32768 Hz external low speed crystal oscillator
HIRC = 12/16 MHz internal high speed RC oscillator
LIRC = 10 kHz internal low speed RC oscillator

Note:

Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.

Figure 6‑8 Clock Controller Block Diagram (3/3)

6.5.4 Functional Description

6.5.4.1 Peripheral clocks

Each peripheral clock has different clock source switching setting. Please refer to the CLKSEL1 and CLKSEL2 description.

	HXT	LXT	HIRC	LIRC	HCLK	PCLK	PLL	Ext. Pin
ADC	Yes	Yes	Yes	-	Yes	-	Yes	-
FDIV	Yes	Yes	Yes	-	Yes	-	-	-
I^2C	-	-	-	-		Yes	-	-
LCD	-	Yes	-	-	-	-	-	-
PWM	Yes	Yes	Yes	-	Yes	-	-	-
RTC	-	Yes	-	-	-	-	-	-
SC	Yes	-	Yes	-	Yes		Yes	-
SPI	-	-	-	-	Yes		Yes	-
TMR	Yes	Yes	Yes	Yes	Yes		-	Yes
UART	Yes	Yes	Yes	-	-		Yes	-
WDT	-	-	-	Yes	-	-	-	-
WWDT	-	-	-	-	Yes	-	-	-

Table 6-7 Peripheral Clocks

6.5.4.2 Clocks in Power-down Mode

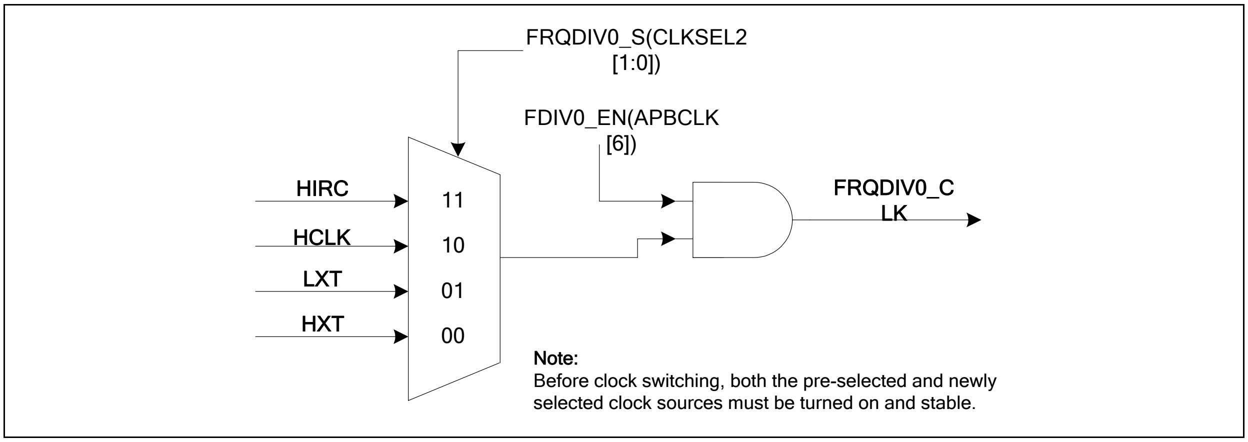
When chip enters Power-down mode, system clocks (CPU clock, HCLKx, and PCLKx), HXT, HIRC will be disabled directly. LXT and LIRC could be still active in Power-down mode if CPU does not disable these clocks before entering Power-down mode. Peripheral module clocks could be still active in Power-down mode if peripheral module adopts LXT or LIRC and LXT or LIRC does not be disabled respectively.

6.5.4.3 Frequency Divider Output

This device is equipped two power-of-2 frequency divide-FDIV0 and FDIV1. Each is composed by16 chained divide-by-2 shift registers. One of the 16 shift register outputs selected by a sixteen to one multiplexer is reflected to IO. Therefore there are 16 options of power-of-2 divided clocks with the frequency from Fin/2^1 to Fin/2^16 where Fin is input clock frequency to the clock divider.

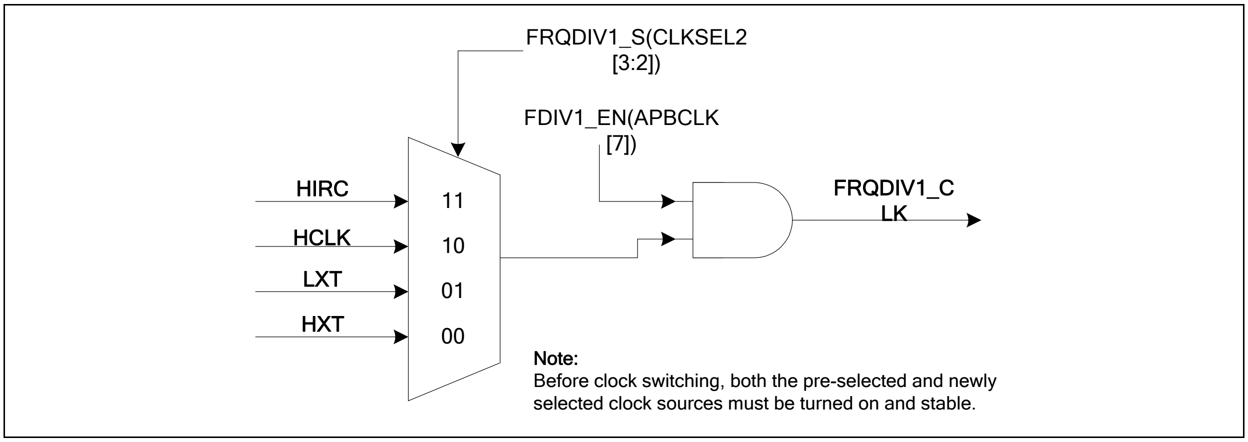
The output formula is Fout = Fin/2^(N+1), where Fin is the input clock frequency, Fout is the clock divider output frequency and N is the 4-bit value in FSEL (FRQDIVx[3:0], x= 0-1).

When FDIV_EN (FRQDIVx[4], x=0-1) is set to high, the rising transition of FDIV_EN will reset the chained counter and then chained counter starts to count. When FDIV_EN (FRQDIVx[4], x= 0-1) is written with zero, the chained counter continuously runs till divided clock reaches low state and stay in low state.



Note:
Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.

Figure 6‑9 Clock Source of Frequency Divider0



Note:
Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.

Figure 6‑10 Clock Source of Frequency Divider1

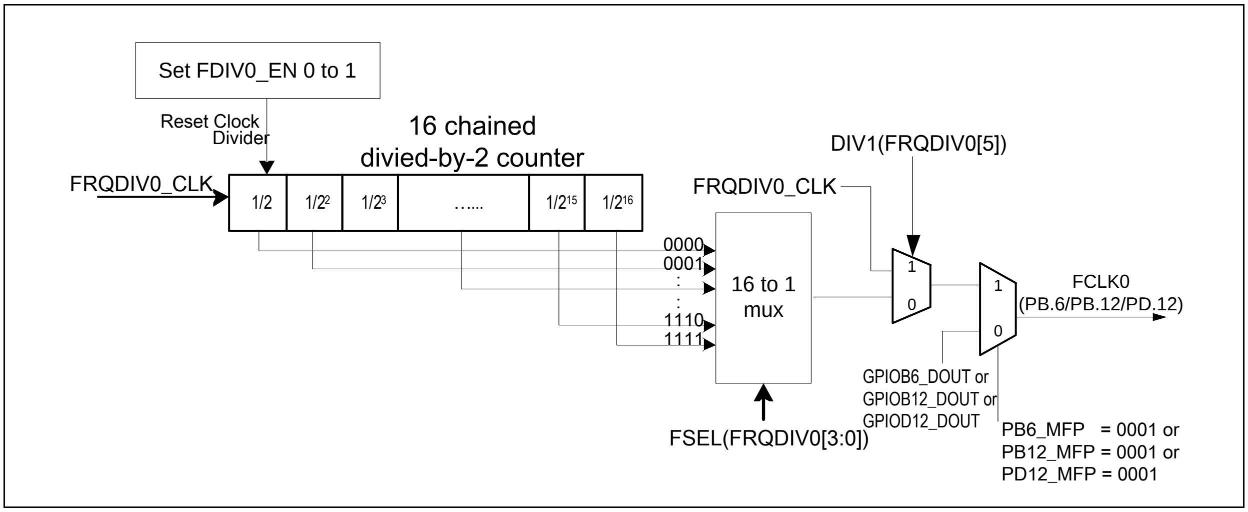


Figure 6‑11 Frequency Divider0 Block Diagram

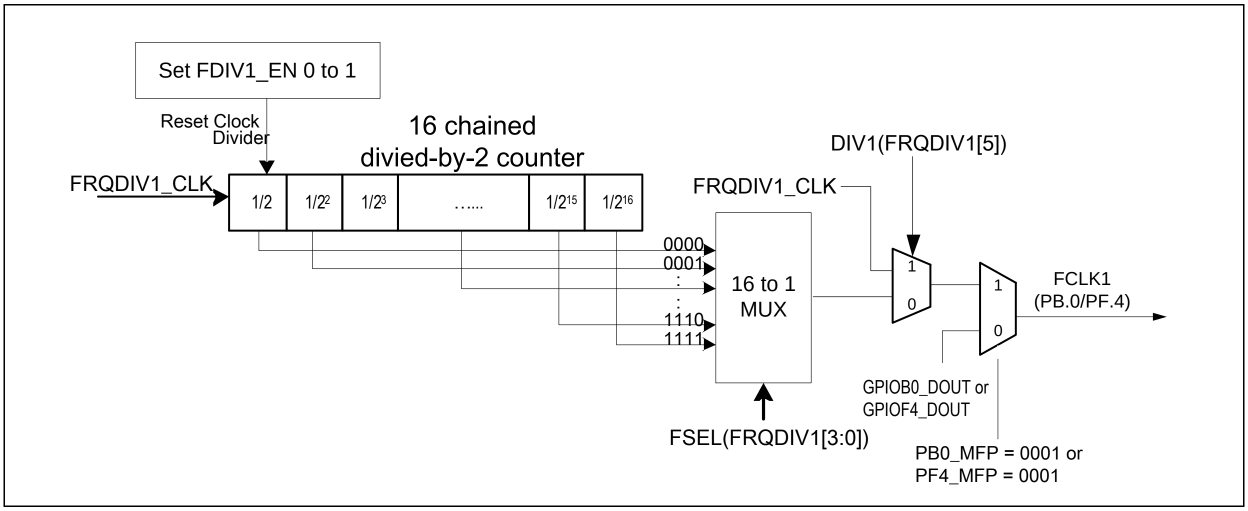


Figure 6-12 Frequency Divider1 Block Diagram

6.5.4.4 PLL

There are two clock sources of PLL: HXT and HIRC. The clock source must be divided into 0.8 ~2.0 MHz first. Then multiple the divided clock from one to 32 times to get PLL output. PLL output must be in range of 16~32 MHz.

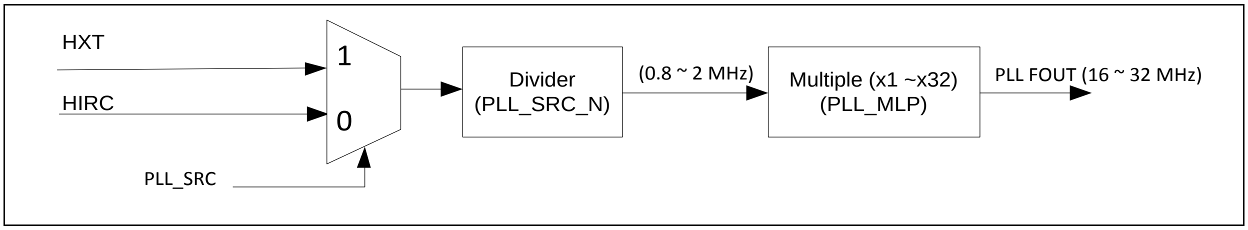


Figure 6-13 PLL Block Diagram

Programming examples:

1. Select PLL input clock source (PLLCTL[17])
2. Divide the input clock source to 0.8~2 MHz (PLLCTL[11:8])
3. Multiple the divided clock to get PLL FOUT (PLLCTL[5:0])

6.5.4.5 HIRC Frequency Changes

There are two output options of HIRC: 12MHz HIRC_FSEL(PWRCTL[12] = 0) or 16MHz HIRC_FSEL(PWRCTL[12] = 1). User can switch HIRC output frequency directly. However there will be transient clock period, whose period is between 12MHz and 16MHz. If user concerns this transient interval, user can uppress HIRC output during this time interval by setting HIRC_F_STOP high.

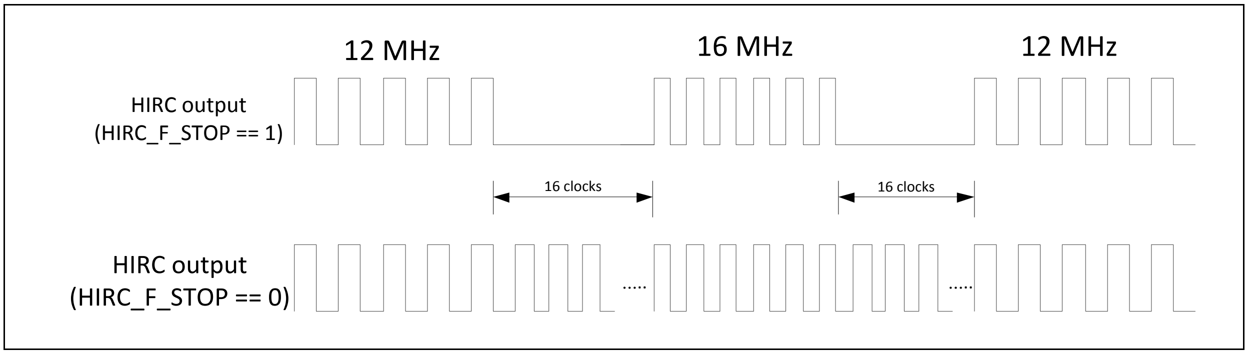


Figure 6-14 HIRC Frequency Change

6.5.5 Register Map

R: read only**, W:** write only**, R/W:** both read and write

Register	**Offset**	**R/W**	**Description**	**Reset Value**
CLK Base Address: **CLK_BA = 0x5000_0200**				
PWRCTL	CLK_BA+0x00	R/W	System Power-down Control Register	0x001A_0B1x
AHBCLK	CLK_BA+0x04	R/W	AHB Devices Clock Enable Control Register	0x0000_0035
APBCLK	CLK_BA+0x08	R/W	APB Devices Clock Enable Control Register	0x0000_0001
CLKSTATUS	CLK_BA+0x0C	R	Clock status monitor Register	0x0000_001x
CLKSEL0	CLK_BA+0x10	R/W	Clock Source Select Control Register 0	0x0000_000x
CLKSEL1	CLK_BA+0x14	R/W	Clock Source Select Control Register 1	0x0018_44F3
CLKSEL2	CLK_BA+0x18	R/W	Clock Source Select Control Register 2	0x0038_440F
CLKDIV0	CLK_BA+0x1C	R/W	Clock Divider Number Register 0	0x0000_0000
CLKDIV1	CLK_BA+0x20	R/W	Clock Divider Number Register 1	0x0000_0000
PLLCTL	CLK_BA+0x24	R/W	PLL Control Register	0x0003_0000
FRQDIV0	CLK_BA+0x28	R/W	Frequency Divider0 Control Register	0x0000_0000
WK_INTSTS	CLK_BA+0x30	R	Wake-up Interrupt Status	0x0000_0000
APB_DIV	CLK_BA+0x34	R/W	APB Clock Divider	0x0000_0000
FRQDIV1	CLK_BA+0x38	R/W	Frequency Divider1 Control Register	0x0000_0000
CLK_SP_DET	CLK_BA+0x3C	R/W	Clock Stop Detect Control Register	0x0000_0000
CLK_SP_STS	CLK_BA+0x40	R	Clock Stop Detect Status Register	0x0000_0000

6.5.6 Register Description

Power-down Control Register （PWRCTL）

Register	Offset	R/W	Description	Reset Value
PWRCTL	CLK_BA+0x00	R/W	System Power-down Control Register	0x001A_0B1x

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		HIRC_F_STOP	HIRC_FSEL	HXT_GAIN		HXT_CUR_SEL	HXT_SELXT
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	PD_EN	PD_WK_IE	WK_DLY	LIRC_EN	HIRC_EN	LXT_EN	HXT_EN

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13]	**HIRC_F_STOP**	**HIRC Stop Output When Frequency Changes** This is a protected register. Please refer to open lock sequence to program it. 0 = HIRC will continue to output when HIRC frequency changes. 1 = HIRC will suppress to output during first 16 clocks when HIRC frequency change.
[12]	**HIRC_FSEL**	**HIRC Output Frequency Select** 0 = HIRC will output 12MHz clock. 1 = HIRC will output 16MHz Clock.
[11:10]	**HXT_GAIN**	**HXT Gain Control Bit** This is a protected register. Please refer to open lock sequence to program it. Gain control is used to enlarge the gain of crystal to make sure crystal wok normally. If gain control is enabled, crystal will consume more power than gain control off. 00 = HXT frequency is lower than from 8 MHz. 01 = HXT frequency is from 8 MHz to 12 MHz. 10 = HXT frequency is from 12 MHz to 16 MHz. 11 = HXT frequency is higher than 16 MHz.
[9]	**HXT_CUR_SEL**	**HXT Internal Current Selection** HXT has some internal current path to help crystal start-up. However when these currnet path existence, HXT will consume more power. User can use this bit to balance the start-up and power consumption. 0 = HXT current path always exists. HXT will consume more power. For 16MHz to 24 MHz crystal. 1 = HXT current path will exist 2ms then cut down. HXT will consume less power. For 4 MHz to 16 MHz crystal.
[8]	**HXT_SELXT**	**HXT SELXT**

Bits	Description	
		This is a protected register. Please refer to open lock sequence to program it. 0 = High frequency crystal loop back path Disabled. It is used for external oscillator. 1 = High frequency crystal loop back path Enabled. It is used for external crystal.
[7]	**Reserved**	Reserved.
[6]	**PD_EN**	**Chip Power-down Mode Enable Bit** This is a protected register. Please refer to open lock sequence to program it. When CPU sets this bit, the chip power down is enabled and chip will not enter Power-down mode until CPU sleep mode is also active. When chip wakes up from Power-down mode, this bit will be auto cleared. When chip is in Power-down mode, the LDO, HXT and HIRC will be disabled, but LXT and LIRC are not controlled by Power-down mode. When power down, the PLL and system clock (CPU, HCLKx and PCLKx) are also disabled no matter the Clock Source selection. Peripheral clocks are not controlled by this bit, if peripheral Clock Source is from LXT or LIRC. In Power-down mode, flash macro power is ON. 0 = Chip operated in Normal mode. 1 = Chip power down Enabled.
[5]	**PD_WK_IE**	**Power-down Mode Wake-up Interrupt Enable Control** This is a protected register. Please refer to open lock sequence to program it. 0 = Disabled. 1 = Enabled. PD_WK_INT will be set if both PD_WK_IS and PD_WK_IE are high.
[4]	**WK_DLY**	**Wake-up Delay Counter Enable Control** This is a protected register. Please refer to open lock sequence to program it. When chip wakes up from Power-down mode, the clock control will delay 4096 clock cycles to wait HXT stable or 16 clock cycles to wait HIRC stable. 0 = Delay clock cycle Disabled. 1 = Delay clock cycle Enabled.
[3]	**LIRC_EN**	**LIRC Enable Control** This is a protected register. Please refer to open lock sequence to program it. 0 = Disabled. 1 = Enabled. LIRC is enabled by default.
[2]	**HIRC_EN**	**HIRC Enable Control** This is a protected register. Please refer to open lock sequence to program it. 0 = Disabled. 1 = Enabled. HIRC is enabled by default.
[1]	**LXT_EN**	**LXT Enable Control** This is a protected register. Please refer to open lock sequence to program it. 0 = Disabled. 1 = Enabled. LXT is disabled by default.
[0]	**HXT_EN**	**HXT Enable Control** This is a protected register. Please refer to open lock sequence to program it. The bit default value is set by flash controller user configuration register config0 [26].

Bits	Description	
		0 = Disabled. 1 = Enabled. HXT is disabled by default.

Mode	PD_EN	CPU Run WFI Instruction	Clock Gating
Normal Mode	0	NO	Depending on S/W setting
Idle Mode (CPU entry sleep mode)	0	YES	Only CPU clock gating
Power-down Mode	1	YES	Most Clocks are gating except LXT or LIRC and IP adopting LXT or LIRC. S/W can turn off LXT and LIRC before chip enters Power-down mode.

Table 6‑8 Power Modes and Clocks

AHB Devices Clock Enable Control Register （AHBCLK）

These register bits are used to enable/disable AHB IP HCLK and engine clocks

Register	Offset	R/W	Description	Reset Value
AHBCLK	CLK_BA+0x04	R/W	AHB Devices Clock Enable Control Register	0x0000_0035

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**TICK_EN**	**SRAM_EN**	**Reserved**	**ISP_EN**	**DMA_EN**	**GPIO_EN**

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**TICK_EN**	**System Tick Clock Enable Control** 0 = Disabled. 1 = Enabled.
[4]	**SRAM_EN**	**SRAM Controller Clock Enable Control** 0 = Disabled. 1 = Enabled.
[3]	**Reserved**	Reserved.
[2]	**ISP_EN**	**Flash ISP Controller Clock Enable Control** 0 = Disabled. 1 = Enabled.
[1]	**DMA_EN**	**DMA Controller Clock Enable Control** 0 = Disabled. 1 = Enabled.
[0]	**GPIO_EN**	**GPIO Controller Clock Enable Control** 0 = Disabled. 1 = Enabled.

APB Devices Clock Enable Control Register （APBCLK）

These register bits are used to enable/disable APB IP PCLK and engine clocks.

Register	Offset	R/W	Description	Reset Value
APBCLK	CLK_BA+0x08	R/W	APB Devices Clock Enable Control Register	0x0000_0001

31	30	29	28	27	26	25	24
SC1_EN	SC0_EN	Reserved	ADC_EN	Reserved	LCD_EN	Reserved	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		PWM0_CH23_EN	PWM0_CH01_EN	Reserved		UART1_EN	UART0_EN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		SPI1_EN	SPI0_EN	ACMP_EN	Reserved	I2C1_EN	I2C0_EN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FDIV1_EN	FDIV0_EN	TMR3_EN	TMR2_EN	TMR1_EN	TMR0_EN	RTC_EN	WDT_EN

Bits	Description	
[31]	**SC1_EN**	**SmartCard 1 Clock Enable Control** 0 = Disabled. 1 = Enabled.
[30]	**SC0_EN**	**SmartCard 0 Clock Enable Control** 0 = Disabled. 1 = Enabled.
[29]	**Reserved**	Reserved.
[28]	**ADC_EN**	**Analog-digital-converter (ADC) Clock Enable Control** 0 = Disabled. 1 = Enabled.
[27]	**Reserved**	Reserved.
[26]	**LCD_EN**	**LCD Controller Clock Enable Control** 0 = Disabled. 1 = Enabled.
[25:22]	**Reserved**	Reserved.
[21]	**PWM0_CH23_EN**	**PWM0 Channel 2 and Channel 3 Clock Enable Control** 0 = Disabled. 1 = Enabled.
[20]	**PWM0_CH01_EN**	**PWM0 Channel 0 and Channel 1Clock Enable Control** 0 = Disabled. 1 = Enabled.
[19:18]	**Reserved**	Reserved.
[17]	**UART1_EN**	**UART1 Clock Enable Control**

Bits	Description	
		0 = Disabled. 1 = Enabled.
[16]	**UART0_EN**	**UART0 Clock Enable Control** 0 = Disabled. 1 = Enabled.
[15:14]	**Reserved**	Reserved.
[13]	**SPI1_EN**	**SPI1 Clock Enable Control** 0 = Disabled. 1 = Enabled.
[12]	**SPI0_EN**	**SPI0 Clock Enable Control** 0 = Disabled. 1 = Enabled.
[11]	**ACMP_EN**	**ACMP Clock Enable Control** 0 = Disabled. 1 = Enabled.
[10]	**Reserved**	Reserved.
[9]	**I2C1_EN**	**I2C1 Clock Enable Control** 0 = Disabled. 1 = Enabled.
[8]	**I2C0_EN**	**I2C0 Clock Enable Control** 0 = Disabled. 1 = Enabled.
[7]	**FDIV1_EN**	**Frequency Divider1 Output Clock Enable Control** 0 = Disabled. 1 = Enabled.
[6]	**FDIV0_EN**	**Frequency Divider0 Output Clock Enable Control** 0 = Disabled. 1 = Enabled.
[5]	**TMR3_EN**	**Timer3 Clock Enable Control** 0 = Disabled. 1 = Enabled.
[4]	**TMR2_EN**	**Timer2 Clock Enable Control** 0 = Disabled. 1 = Enabled.
[3]	**TMR1_EN**	**Timer1 Clock Enable Control** 0 = Disabled. 1 = Enabled.
[2]	**TMR0_EN**	**Timer0 Clock Enable Control** 0 = Disabled. 1 = Enabled.
[1]	**RTC_EN**	**Real-time-clock Clock Enable Control** This bit is used to control the RTC APB clock only, The RTC engine Clock Source is from

Bits	Description	
		LXT. 0 = Disabled. 1 = Enabled.
[0]	**WDT_EN**	**Watchdog Timer Clock Enable Control** This is a protected register. Please refer to open lock sequence to program it. This bit is used to control the WDT APB clock only, The WDT engine Clock Source is from LIRC. 0 = Disabled. 1 = Enabled.

Clock status Register （CLKSTATUS）

These register bits are used to monitor if the chip Clock Source stable or not, and if clock switch is fail

Register	Offset	R/W	Description	Reset Value
CLKSTATUS	CLK_BA+0x0C	R	Clock status monitor Register	0x0000_001x

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CLK_SW_FAIL	**Reserved**		**HIRC_STB**	**LIRC_STB**	**PLL_STB**	**LXT_STB**	**HXT_STB**

Bits	Description	
[31:8]		Reserved.
[7]	**CLK_SW_FAIL**	**Clock Switch Fail Flag** 0 = Clock switch success. 1 = Clock switch fail. This bit will be set when target switch Clock Source is not stable. This bit is write 1 clear
[6:5]	**Reserved**	Reserved.
[4]	**HIRC_STB**	**HIRC Clock Source Stable Flag** 0 = HIRC clock is not stable or not enable. 1 = HIRC clock is stable.
[3]	**LIRC_STB**	**LIRC Clock Source Stable Flag** 0 = LIRC clock is not stable or not enable. 1 = LIRC clock is stable.
[2]	**PLL_STB**	**PLL Clock Source Stable Flag** 0 = PLL clock is not stable or not enable. 1 = PLL clock is stable.
[1]	**LXT_STB**	**LXT Clock Source Stable Flag** 0 = LXT clock is not stable or not enable. 1 = LXT clock is stable.
[0]	**HXT_STB**	**HXT Clock Source Stable Flag** 0 = HXT clock is not stable or not enable. 1 = HXT clock is stable.

Clock Source Select Control Register 0 （CLKSEL0）

Register	Offset	R/W	Description	Reset Value
CLKSEL0	CLK_BA+0x10	R/W	Clock Source Select Control Register 0	0x0000_000x

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="5">Reserved</td><td colspan="3">HCLK_S</td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:3]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
<tr><td>[2:0]</td><td><b>HCLK_S</b></td><td><b>HCLK Clock Source Selection</b>
This is a protected register. Please refer to open lock sequence to program it.
<b>Note:</b>
Before Clock Source switches, the related clock sources (pre-select and new-select) must be turn on
The 3-bit default value is reloaded with the value of CFOSC (Config0[26:24]) in user configuration register in Flash controller by any reset. Therefore the default value is either 000b or 111b.
<table>
<tr><th>HCLK_S</th><th>Clock Source</th></tr>
<tr><td>000</td><td>HXT</td></tr>
<tr><td>001</td><td>LXT</td></tr>
<tr><td>010</td><td>PLL clock</td></tr>
<tr><td>011</td><td>LIRC</td></tr>
<tr><td>111</td><td>HIRC</td></tr>
</table></td></tr>
</table>

Clock Source Select Control Register 1 （CLKSEL1）

Register	Offset	R/W	Description	Reset Value
CLKSEL1	CLK_BA+0x14	R/W	Clock Source Select Control Register 1	0x0018_44F3

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		ADC_S			LCD_S	Reserved	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	TMR1_S			Reserved	TMR0_S		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PWM0_CH23_S		PWM0_CH01_S		Reserved		UART_S	

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:22]</td><td></td><td>Reserved.</td></tr>
<tr><td>[21:19]</td><td><b>ADC_S</b></td><td><b>ADC Clock Source Selection</b>
<table>
<tr><th>ADC_S</th><th>Clock Source</th></tr>
<tr><td>000</td><td>HXT</td></tr>
<tr><td>001</td><td>LXT</td></tr>
<tr><td>010</td><td>PLL clock</td></tr>
<tr><td>011</td><td>HIRC</td></tr>
<tr><td>others</td><td>HCLK</td></tr>
</table></td></tr>
<tr><td>[18]</td><td><b>LCD_S</b></td><td><b>LCD Clock Source Selection</b>
0 = Clock Source from LXT.
1 = Reserved.</td></tr>
<tr><td>[17:15]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
<tr><td>[14:12]</td><td><b>TMR1_S</b></td><td><b>Timer1 Clock Source Selection</b>
<table>
<tr><th>TMR1_S</th><th>Clock Source</th></tr>
<tr><td>000</td><td>HXT</td></tr>
<tr><td>001</td><td>LXT</td></tr>
<tr><td>010</td><td>LIRC</td></tr>
<tr><td>011</td><td>external pin</td></tr>
<tr><td>100</td><td>HIRC</td></tr>
<tr><td>others</td><td>HCLK</td></tr>
</table></td></tr>
<tr><td>[11]</td><td></td><td>Reserved.</td></tr>
<tr><td>[10:8]</td><td><b>TMR0_S</b></td><td><b>Timer0 Clock Source Selection</b></td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>TMR0_S</th><th>Clock Source</th></tr>
<tr><td>000</td><td>HXT</td></tr>
<tr><td>001</td><td>LXT</td></tr>
<tr><td>010</td><td>LIRC</td></tr>
<tr><td>011</td><td>external pin</td></tr>
<tr><td>100</td><td>HIRC</td></tr>
<tr><td>others</td><td>HCLK</td></tr>
</table>
</td></tr>
<tr><td>[7:6]</td><td><b>PWM0_CH23_S</b></td><td><b>PWM0 Channel 2 and Channel 3 Clock Source Selection</b>
PWM0 channel 2 and channel 3 use the same Engine clock source, both of them with the same prescaler
<table>
<tr><th>PWM0_CH23_S</th><th>Clock Source</th></tr>
<tr><td>00</td><td>HXT</td></tr>
<tr><td>01</td><td>LXT</td></tr>
<tr><td>10</td><td>HCLK</td></tr>
<tr><td>11</td><td>HIRC</td></tr>
</table>
</td></tr>
<tr><td>[5:4]</td><td><b>PWM0_CH01_S</b></td><td><b>PWM0 Channel 0 and Channel 1 Clock Source Selection</b>
PWM0 channel 0 and channel 1 use the same Engine clock source, both of them with the same prescaler
<table>
<tr><th>PWM0_CH01_S</th><th>Clock Source</th></tr>
<tr><td>00</td><td>HXT</td></tr>
<tr><td>01</td><td>LXT</td></tr>
<tr><td>10</td><td>HCLK</td></tr>
<tr><td>11</td><td>HIRC</td></tr>
</table>
</td></tr>
<tr><td>[3:2]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
<tr><td>[1:0]</td><td><b>UART_S</b></td><td><b>UART 0/1 Clock Source Selection (UART0 and UART1 Use the Same Clock Source Selection)</b>
<table>
<tr><th>UART_S</th><th>Clock Source</th></tr>
<tr><td>00</td><td>HXT</td></tr>
<tr><td>01</td><td>LXT</td></tr>
<tr><td>10</td><td>PLL clock</td></tr>
<tr><td>11</td><td>HIRC</td></tr>
</table>
</td></tr>
</table>

Clock Source Select Control Register 2 （CLKSEL2）

Before clock switch the related clock sources (pre-select and new-select) must be turned on.

Register	Offset	R/W	Description	Reset Value
CLKSEL2	CLK_BA+0x18	R/W	Clock Source Select Control Register 2	0x0038_440F

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		SPI1_S	SPI0_S	SC_S		Reserved	
15	14	13	12	11	10	9	8
Reserved	TMR3_S			Reserved	TMR2_S		
7	6	5	4	3	2	1	0
Reserved				FRQDIV0_S		FRQDIV1_S	

Bits	Description	
[31:22]	Reserved	Reserved.
[21]	SPI1_S	**SPI1 Clock Source Selection** **0 = PLL.** **1 = HCLK.**
[20]	SPI0_S	**SPI0 Clock Source Selection** **0 = PLL.** **1 = HCLK.**
[19:18]	SC_S	**SC Clock Source Selection** SC_S / Clock Source: 00 = HXT 01 = PLL output 10 = HIRC 11 = HCLK
[17:15]		Reserved.
[14:12]	TMR3_S	**Timer3 Clock Source Selection** TMR3_S / Clock Source: 000 = HXT 001 = LXT 010 = LIRC 011 = external pin 100 = HIRC

Bits	Description		
		others	HCLK
[11]	**Reserved**	Reserved.	
[10:8]	**TMR2_S**	**Timer2 Clock Source Selection**	
		TMR2_S	**Clock Source**
		000	HXT
		001	LXT
		010	LIRC
		011	external pin
		100	HIRC
		others	HCLK
[7:4]	**Reserved**	Reserved.	
[3:2]	**FRQDIV0_S**	**Clock Divider0 Clock Source Selection**	
		FRQDIV_S	**Clock Source**
		00	HXT
		01	LXT
		10	HCLK
		11	HIRC
[1:0]	**FRQDIV1_S**	**Clock Divider Clock1 Source Selection**	
		FRQDIV_S	**Clock Source**
		00	HXT
		01	LXT
		10	HCLK
		11	HIRC

Clock Divider 0 Register (CLKDIV0)

Register	Offset	R/W	Description	Reset Value
CLKDIV0	CLK_BA+0x1C	R/W	Clock Divider Number Register 0	0x0000_0000

31	30	29	28	27	26	25	24
SC0_N				**Reserved**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ADC_N							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**UART_N**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**HCLK_N**			

Bits	Description	
[31:28]	**SC0_N**	**SC 0 Clock Divide Number From SC 0 Clock Source** The SC 0 clock frequency = (SC0 Clock Source frequency) / (SC0_N + 1).
[27:24]	**Reserved**	Reserved.
[23:16]	**ADC_N**	**ADC Clock Divide Number From ADC Clock Source** The ADC clock frequency = (ADC Clock Source frequency) / (ADC_N + 1).
[15:12]	**Reserved**	Reserved.
[11:8]	**UART_N**	**UART Clock Divide Number From UART Clock Source** The UART clock frequency = (UART Clock Source frequency) / (UART_N + 1).
[7:4]	**Reserved**	Reserved.
[3:0]	**HCLK_N**	**HCLK Clock Divide Number From HCLK Clock Source** The HCLK clock frequency = (HCLK Clock Source frequency) / (HCLK_N + 1).

Clock Divider 1 Register (CLKDIV1)

Register	Offset	R/W	Description	Reset Value
CLKDIV1	CLK_BA+0x20	R/W	Clock Divider Number Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TMR3_N				**TMR2_N**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TMR1_N				**TMR0_N**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**SC1_N**			

Bits	Description	
[31:24]		Reserved.
[23:20]	**TMR3_N**	**Timer3 Clock Divide Number From Timer3 Clock Source** The Timer3 clock frequency = (Timer3 Clock Source frequency) / (TMR3_N + 1).
[19:16]	**TMR2_N**	**Timer2 Clock Divide Number From Timer2 Clock Source** The Timer2 clock frequency = (Timer2 Clock Source frequency) / (TMR2_N + 1).
[15:12]	**TMR1_N**	**Timer1 Clock Divide Number From Timer1 Clock Source** The Timer1 clock frequency = (Timer1 Clock Source frequency) / (TMR1_N + 1).
[11:8]	**TMR0_N**	**Timer0 Clock Divide Number From Timer0 Clock Source** The Timer0 clock frequency = (Timer0 Clock Source frequency) / (TMR0_N + 1).
[7:4]	**Reserved**	Reserved.
[3:0]	**SC1_N**	**SC 1 Clock Divide Number From SC 1 Clock Source** The SC 1 clock frequency = (SC 1 Clock Source frequency) / (SC1_N + 1).

PLL Control Register （PLLCTL）

The PLL reference clock input is from HXT or HIRC. This register is used to control PLL output frequency and PLL operating mode

Register	Offset	R/W	Description	Reset Value
PLLCTL	CLK_BA+0x24	R/W	PLL Control Register	0x0003_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						PLL_SRC	PD
15	14	13	12	11	10	9	8
Reserved				PLL_SRC_N			
7	6	5	4	3	2	1	0
Reserved		PLL_MLP					

Bits	Description	
[31:18]	Reserved	Reserved.
[17]	PLL_SRC	**PLL Source Clock Select** 0 = PLL source clock from HXT. 1 = PLL source clock from HIRC.
[16]	PD	**Power-down Mode** If set the PD_EN bit "1" in PWR_CTL register, the PLL will enter Power-down mode too 0 = PLL is in normal mode. 1 = PLL is in Power-down mode (default).
[15:12]	Reserved	Reserved.
[11:8]	PLL_SRC_N	**PLL Input Source Divider** The PLL input clock frequency = (PLL Clock Source frequency) / (PLL_SRC_N + 1). PLL input clock frequency range: 0.8MHz ~ 2MHz
[7:6]	Reserved	Reserved.
[5:0]	PLL_MLP	**PLL Multiple** **000000: Reserved** **000001: X1** **000010: X2** **000011: X3** **000100: X4** **…** 010000:X16 … 100000: X32 0thers: Reserved

		PLL output frequency: PLL input frequency * PLL_MLP. PLL output frequency range: 16MHz ~ 32MHz

Frequency Divider Control Register (FRQDIV0)

Register	Offset	R/W	Description	Reset Value
FRQDIV0	CLK_BA+0x28	R/W	Frequency Divider0 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**DIV1**	**FDIV_EN**	**FSEL**			

Bits	Description	
[31:5]	**Reserved**	Reserved.
[5]	**DIV1**	**Output Frequency Divied by 1** 0 = Output frequency is equal to FCLK0. 1 = Output frequency is equal to FRQDIV0_CLK.
[4]	**FDIV_EN**	**Frequency Divider Enable Bit** 0 = Frequency Divider Disabled. 1 = Frequency Divider Enabled.
[3:0]	**FSEL**	**Divider Output Frequency Selection Bits** The formula of output frequency is FCLK0 = FRQDIV0_CLK/2^(N+1),. Where FRQDIV0_CLK is the input clock frequency, Fout is the frequency of divider output clock and N is the 4-bit value of FSEL[3:0].

Wake-up Interrupts Status (WK_INTSTS)

Register	Offset	R/W	Description	Reset Value
WK_INTSTS	CLK_BA+0x30	R	Wake-up Interrupt Status	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**PD_WK_IS**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**PD_WK_IS**	**Wake-up Interrupt Sstatus in Chip Power-down Mode** This bit indicates that some event resumes chip from Power-down mode The status is set if external interrupts, UART, GPIO, RTC, USB, SPI, Timer, WDT, and BOD wake-up occurred. Write 1 to clear this bit.

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

APB Clock Divider Register (APB_DIV)

Register	Offset	R/W	Description	Reset Value
APB_DIV	CLK_BA+0x34	R/W	APB Clock Divider	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					APBDIV		

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**APBDIV**	**APB Clock Divider** APB PCLK can be divided from HCLK. 000: PCLK = HCLK. 001: PCLK =1/2 HCLK. 010: PCLK = 1/4 HCLK. 011: PCLK = 1/8 HCLK. 100: PCLK = 1/16 HCLK. Others: PCLK = HCLK.

Frequency Divider1 Control Register (FRQDIV1)

Register	Offset	R/W	Description	Reset Value
FRQDIV1	CLK_BA+0x38	R/W	Frequency Divider1 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**DIV1**	**FDIV_EN**	**FSEL**			

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**DIV1**	**Output Frequency Divied by 1** 0 = Output frequency is equal to FCLK1. 1 = Output frequency is equal to FRQDIV1_CLK.
[4]	**FDIV_EN**	**Frequency Divider Enable Bit** 0 = Frequency Divider Disabled. 1 = Frequency Divider Enabled.
[3:0]	**FSEL**	**Divider Output Frequency Selection Bits** The formula of output frequency is FCLK1 = FRQDIV1_CLK /2^(N+1),. Where FRQDIV1_CLK is the input clock frequency, Fout is the frequency of divider output clock and N is the 4-bit value of FSEL[3:0].

Clock Stop Detect Control Register (CLK_SP_DET)

Register	Offset	R/W	Description	Reset Value
CLK_SP_DET	CLK_BA+0x3C	R/W	Clock Stop Detect Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		HIRC_STOP_IE	HIRC_DET	HXT_STOP_IE	HXT_DET	HCLK_DET_IE	HCLK_DET

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**HIRC_STOP_IE**	**HIRC Stop Detect Interrupt Enable Control** 0 = HIRC stop detect interrupt Disabled. 1 = HIRC stop detect interrupt Enabled.
[4]	**HIRC_DET**	**HIRC Stop Detect Enable Control** 0 = HIRC stop detect Disabled. 1 = HIRC stop detect Enabled.
[3]	**HXT_STOP_IE**	**HXT Stop Detect Interrupt Enable Control** 0 = HXT stop detect interrupt Disabled. 1 = HXT stop detect interrupt Enabled.
[2]	**HXT_DET**	**HXT Stop Detect Enable Control** 0 = HXT stop detect Disabled. 1 = HXT stop detect Enabled.
[1]	**HCLK_DET_IE**	**HCLK Stop Detect Interrupt Enable Control** 0 = HCLK stop detect interrupt Disabled. 1 = HCLK stop detect interrupt Enabled.
[0]	**HCLK_DET**	**HCLK Stop Detect Enable Control** 0 = HCLK stop detect Disabled. 1 = HCLK stop detect Enabled. Once HCLK stop detected, hardware will force HCLK from LIRC.

Clock Stop Detect Status Register (CLK_SP_STS)

Register	Offset	R/W	Description	Reset Value
CLK_SP_STS	CLK_BA+0x40	R	Clock Stop Detect Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**H_TCLK_SEL**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**HIRC_SP_IS**	**Reserved**	**HXT_SP_IS**	**Reserved**	**HCLK_SP_IS**

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:11]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
<tr><td>[10:8]</td><td><b>H_TCLK_SEL</b></td><td><b>HCLK Target Clock Select</b>
<table>
<tr><th>H_TCLK_SEL</th><th>Clock Source</th></tr>
<tr><td>000</td><td>HXT</td></tr>
<tr><td>001</td><td>LXT</td></tr>
<tr><td>010</td><td>PLL clock</td></tr>
<tr><td>011</td><td>LIRC</td></tr>
<tr><td>111</td><td>HIRC</td></tr>
</table></td></tr>
<tr><td>[7:5]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
<tr><td>[4]</td><td><b>HIRC_SP_IS</b></td><td><b>HIRC Stop Flag</b>
0 = HIRC normal.
1 = HIRC abnormal.</td></tr>
<tr><td>[3]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
<tr><td>[2]</td><td><b>HXT_SP_IS</b></td><td><b>HXT Stop Flag</b>
0 = HXT normal.
1 = HXT abnormal.</td></tr>
<tr><td>[1]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
<tr><td>[0]</td><td><b>HCLK_SP_IS</b></td><td><b>HCLK Clock Stop Flag</b>
0 = HCLK normal.
1 = HCLK abnormal.</td></tr>
</table>

6.6 Flash Memory Controller (FMC)

6.6.1 Overview

This chip is equipped with 16/32 Kbytes on-chip embedded flash memory for application program memory (APROM) that can be updated through ISP/IAP procedure. In System Programming (ISP) function enables user to update program memory when chip is soldered on PCB. After chip powered on Cortex-M0 CPU fetches code from APROM or LDROM decided by boot select (CBS) in Config0. By the way, this chip also provides Data Flash Region, the Data Flash is shared with original program memory and its start address is configurable and defined by user in Config1. The Data Flash size is defined by user application request.

6.6.2 Features

- 16/32 Kbytes application program memory (APROM)
- 4 Kbytes in system programming (ISP) loader program memory (LDROM)
- Programmable Data Flash start address and memory size with 512 bytes page erase unit
- 512 bytes system program memory (SPROM)
- In System Program (ISP)/In Application Program (IAP) to update on chip flash memory

6.6.3 Block Diagram

The flash memory controller consists of AHB slave interface, ISP control logic, writer interface and flash macro interface timing control logic. The block diagram of flash memory controller is shown as follows.

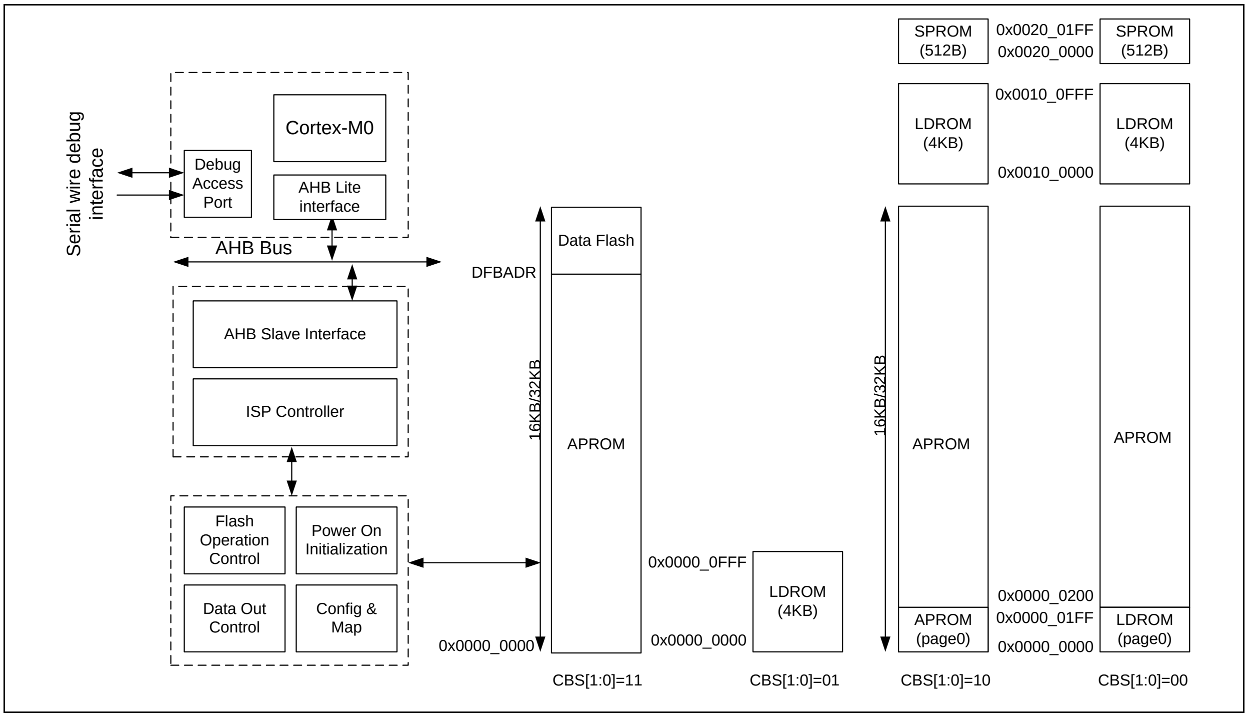


Figure 6−15 Flash Memory Controller

6.6.4 Functional Description

6.6.4.1 *Flash Memory Organization*

The flash memory consists of application program memory (16KB/32KB), Data Flash, ISP loader program memory, and user configuration. The user configuration block provides several bytes to control system logic, such as flash security lock, boot select, Brown-out voltage level, and Data Flash base address. It works like a fuse for power on setting. It is loaded from flash memory to its corresponding control registers during chip powered on. User can set these bits according to application request by writer before chip is mounted on PCB. The data flash start address and its size can defined by user application.

Block Name	DFEN	Size	Start Address	End Address
APROM	0	(16-0.5*N) Kbytes (32-0.5*N) Kbytes	0x0000_0000	DFBADR-1
APROM	1	16 Kbytes 32 Kbytes	0x0000_0000	0x0000_3FFF 0x0000_7FFF
Data Flash	0	0.5*N Kbytes	DFBADR	0x0000_3FFF

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

				0x0000_7FFF
Data Flash	1	N/A	N/A	N/A
LDROM	x	4 Kbytes	0x0010_0000	0x0010_0FFF
SPROM	x	512 bytes	0x0020_0000	0x0020_01FF
User Configuration	x	2 words	0x0030_0000	0x0030_0004

Table 6-9 Memory Access Map

The Flash memory organization is shown below.

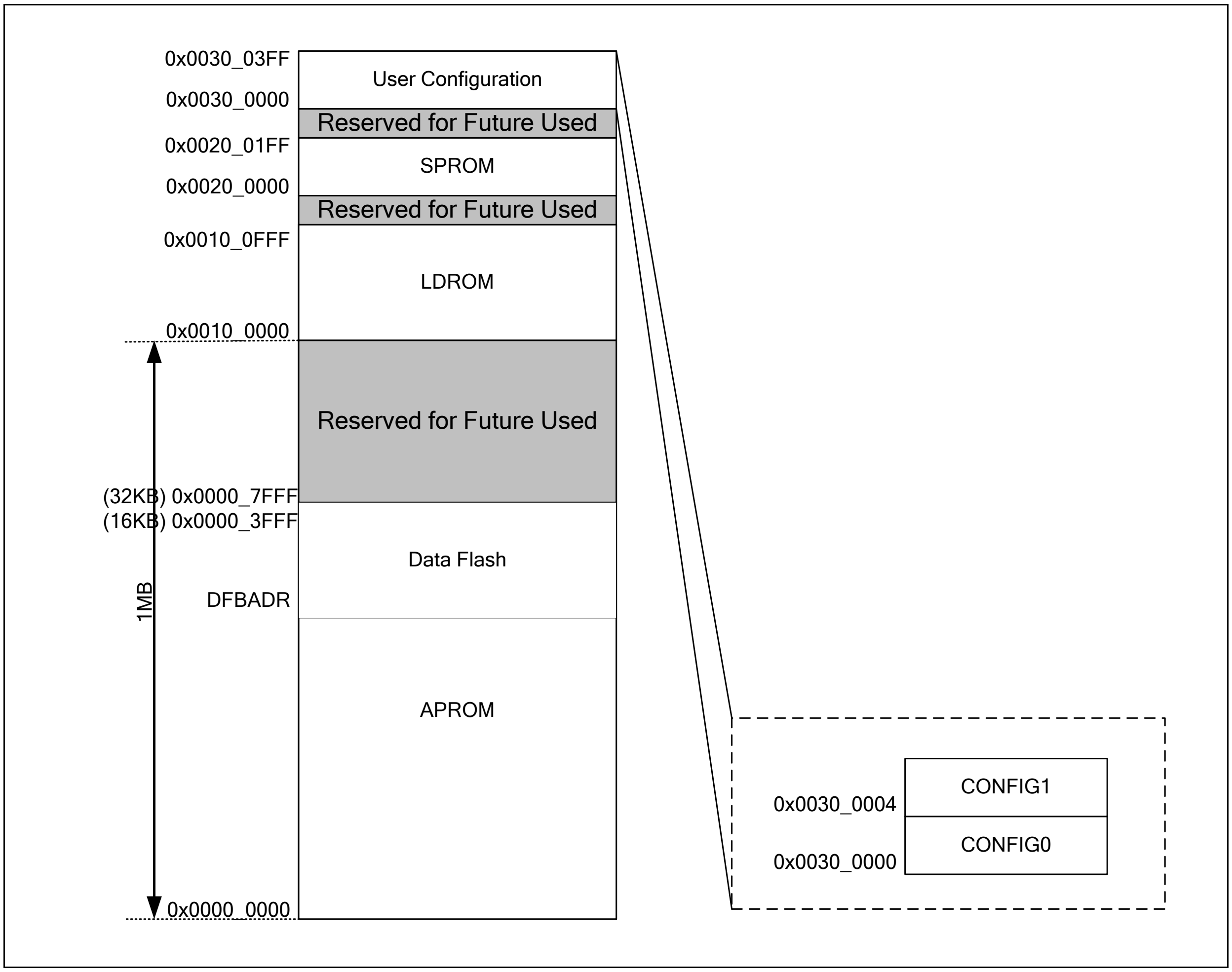


Figure 6-16 Flash Memory Organization

6.6.4.2 Boot Selection

This chip provides in system programming (ISP) feature for user to update program memory when chip is mounted on PCB. A dedicated 4 Kbytes program memory (LDROM) is used to store ISP firmware. Users can select to start program from APROM or LDROM by (CBS) in Config0.

CBS[1:0]	Boot Selection
00	**LDROM with IAP function** Chip booting from LDROM, program executing range including SPROM, LDROM and APROM (except APROM's first page). LDROM address is mapping to 0x0010_0000 ~ 0x0010_0FFF, and the first 512 bytes of LDROM is mapping to the address 0x0000_0000 ~ 0x0000_01FF at the same time. Both APROM and LDROM are programmable in this mode no matter the code is currently running on LDROM or APROM. Data Flash is not functioned in this mode.
01	**LDROM without IAP function** Chip booting from LDROM, program executing range only including SPROM and LDROM. APROM can only be accessed by ISP commands. LDROM is write-protected in this mode.
10	**APROM with IAP function** Chip booting from APROM, program executing range including SPROM, LDROM and APROM. LDROM address is mapping to 0x0010_0000~0x0010_0FFF. Both APROM and LDROM are programmable in this mode no matter the code is currently running on LDROM or APROM. Data Flash is not functioned in this mode.
11	**APROM without IAP function** Chip booting from APROM and program executing range only including SPROM and APROM. LDROM can only be access by ISP commands. APROM is write-protected in this mode.

Table 6‑10 Boot Selection

CBS[1:0]	Boot From	Vector Re-Map	Run In LDROM Write To APROM	Run In APROM Write To LDROM	Run In LDROM Write To LDROM	Run In APROM Write To APROM
00	LDROM	Yes	Yes	-	Yes	-
01	LDROM	-	Yes	-	Yes	-
10	APROM	Yes	-	Yes	-	Yes
11	APROM	-	-	Yes	-	Yes

Table 6‑11 Boot Selection and Supports Function

6.6.4.3 BOD

The brown-out detection function is used for monitoring the voltage on V_{DD} pin. If V_{DD} voltage falls below level setting of CBOV, the BOD event will be triggered when BOD enabled. User can decide to use BOD reset by enable CBORST or just enable BOD interrupt (set CBORST= 11) when BOD detected. Because BOD reset is issued whenever V_{DD} voltage falls below the level setting of CBOV, user must make sure the CBOV setting to avoid BOD reset right after BOD reset enabled. For example, if the V_{DD} is 2.2V, CBOV could only be 10. Otherwise, the system will be halted in BOD reset state when CBOV is 01 or 00.

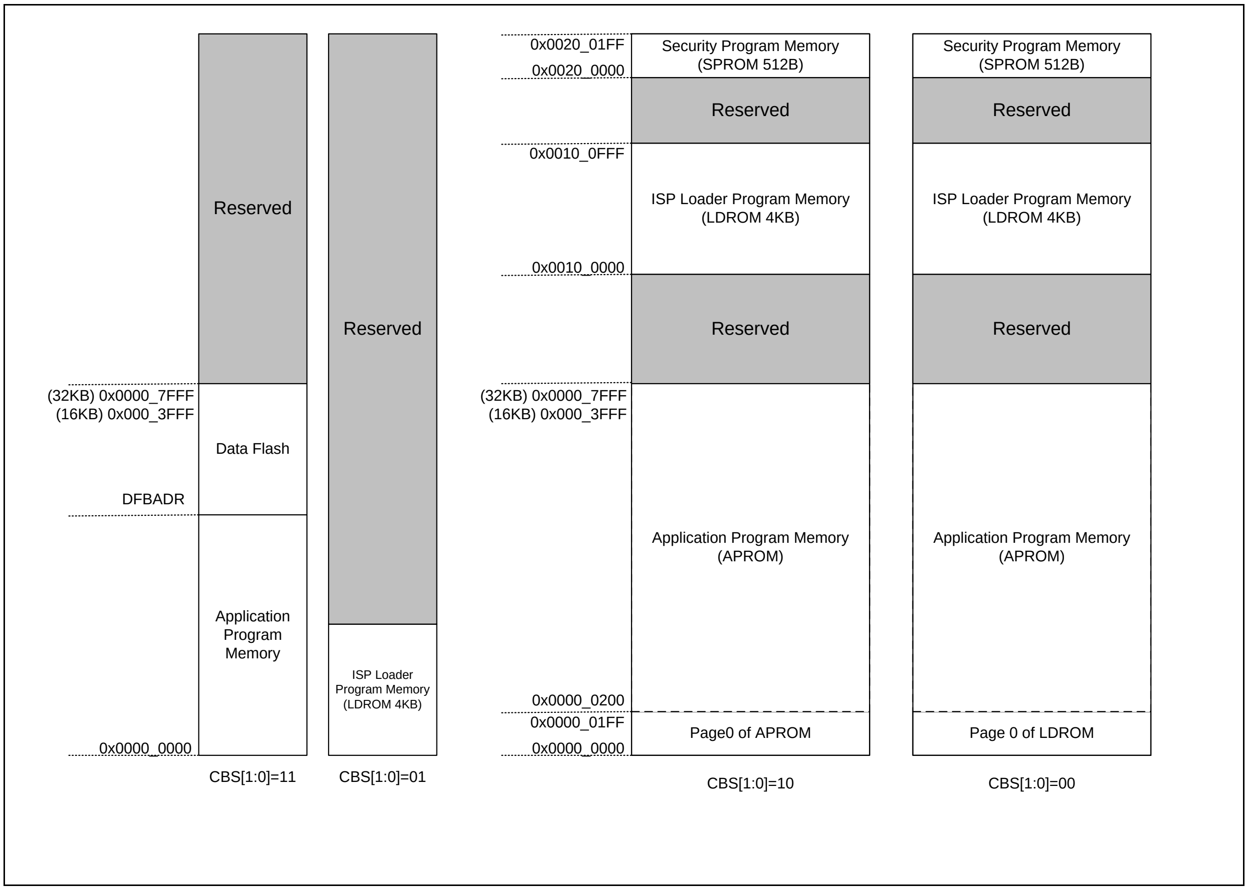


Figure 6‑17 Flash Memory Mapping of CBS in CONFIG0

6.6.4.4 In Application Programming

This chip provides In-application-programming (IAP) function for user to switch the code executing between APROM and LDROM without a reset. User can enable the IAP function by re-booting chip and setting the chip boot selection bits in CONFIG0 (CBS[1:0]) as ‘10’b or ‘00’b.

In the case that chip boots from APROM with the IAP function enabled (CBS[1:0] = ‘10’b), the executable range of code includes all of APROM and LDROM. The address space of APROM is kept as the original size but the address space of the 4 KB LDROM is mapped to 0x0010_0000~ 0x0010_0FFF.

In the case that chip boots from LDROM with the IAP function enabled (CBS[1:0] = ‘00’b), the executable range of code includes all of LDROM and almost all of APROM except for its first page. User cannot access the first page of APROM because the first page of executable code range becomes the mirror of the first page of LDROM by default. Meanwhile, the address space of 4 KB LDROM is mapped to 0x0010_0000~0x0010_0FFF.

Please refer to the following figure for the address map while IAP is activating.

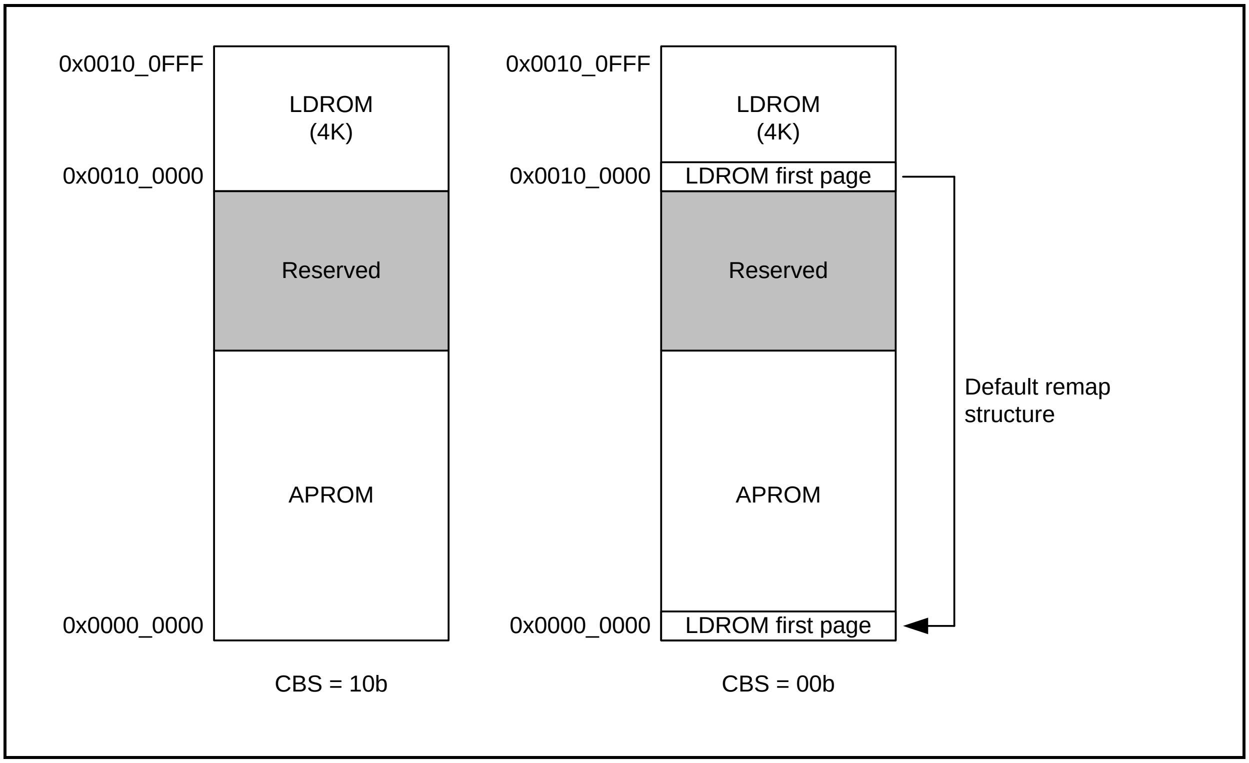


Figure 6-18 Executable Range of Code with IAP Function Enabled

When chip boots with the IAP function enabled, any other page within the executable range of code can be mirrored to the first page of executable code (0x0000_0000~0x0000_07FF) any time. User can change the remap address of the first executing page by filling the target remap address to ISPADR and then go through ISP procedure with the Vector Page Re-map command. After changing the remap address, user can check if the change is successful by reading the VECMAP field in the ISPSTA register.

6.6.4.5 Data Flash

This chip provides Data Flash for user to store data. It is read/written through ISP commands. The size of each erase unit is 512 bytes. When a word will be changed, all 128 words need to be copied to another page or SRAM in advance. The Data Flash base address is defined by DFBADR if DFEN bit in Config0 is enabled and application program memory size is (16-0.5*N) Kbytes for 16 Kbytes flash, (32-0.5*N) Kbytes for 32 Kbytes flash and Data Flash size is 0.5*N Kbytes.

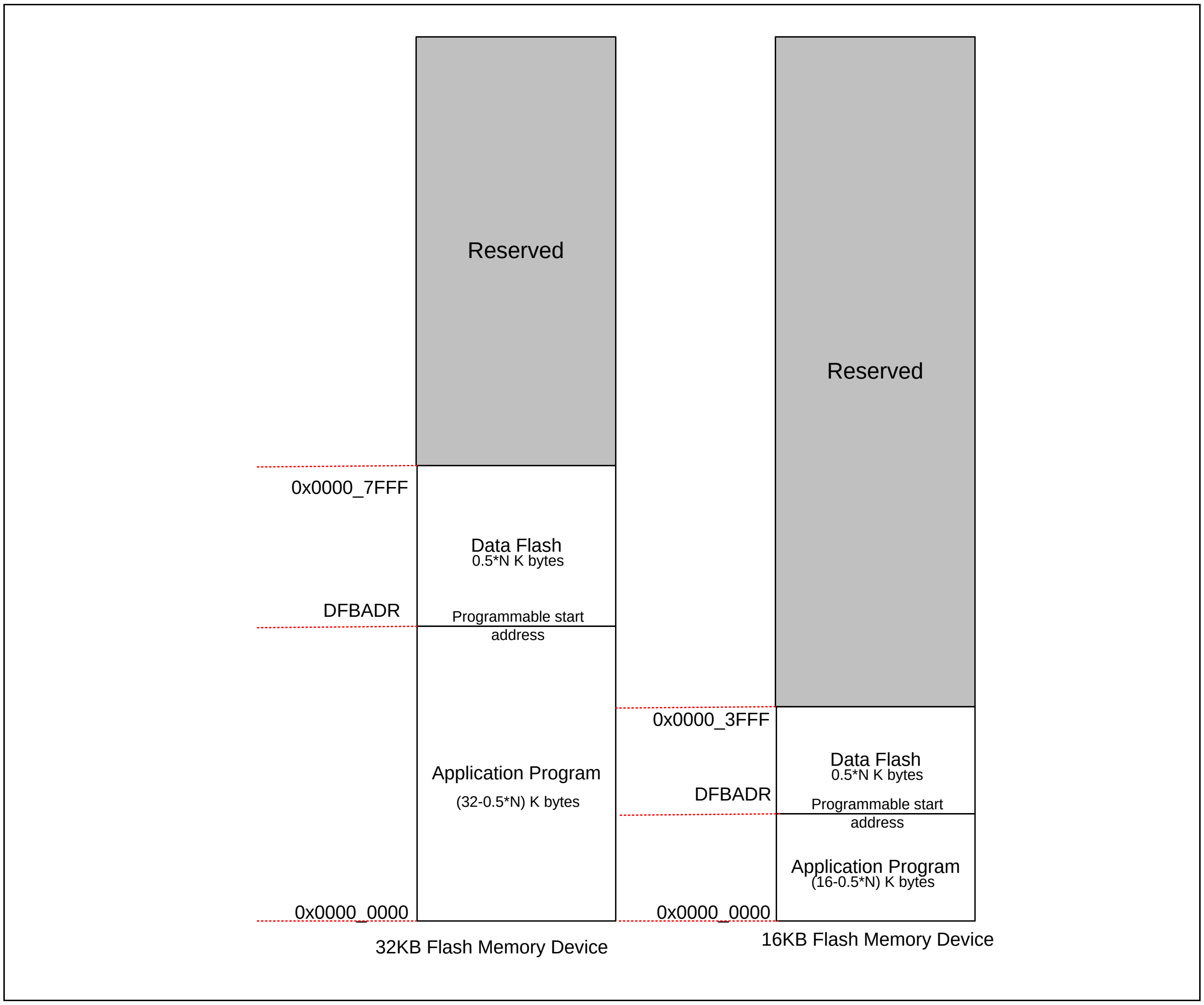


Figure 6‑19 16/32KB Flash Memory Structure

6.6.4.6 Security Program Memory (SPROM)

This chip provides security program memory for user to store instruction for security application. It is read/written through ISP procedure and ICE, and this memory cannot be erased by “whole chip erase command” but “page erase command”. The security program memory is 512 bytes with location address 0x20_0000 ~ 0x20_01FF. The last byte of this memory is used to identify the code is secured or non-secured. Refer the following table, it shows that security program memory only allows CPU performs instruction fetch and page-erase operation when it is secured code.

- (The last byte= 0xFF): Non-secured code.

	ICE Debug	ISP/IAP	CPU Data	CPU Instruction
Whole chip erase	-	-	-	-
Page-erase	-	O	-	-
Program	-	O	-	-
Read	O	O	O	O

- (The last byte=Others): Secured code

	ICE Debug	ISP/IAP	CPU Data	CPU Instruction
Whole chip erase	-	-	-	-
Page-erase	-	O	-	-
Program	-	X	-	-
Read	FFh	FFh	FFh	CPU Instruction

6.6.4.7 User Configuration

CONFIG0 (Address = 0x0030_0000)

31	30	29	28	27	26	25	24
Reserved					CFOSC	Reserved	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			CBORST		Reserved		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CHXT_GAIN			Reserved				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CBS		Reserved				LOCK	DFEN

CONFIG0	Address = 0x0030_0000	
Bits	**Description**	
[31]	Reserved	Should be set to 1
[30:27]	Reserved	Reserved
[26]	CFOSC	**CPU Clock Source Selection After Reset** CFOSC | Clock Source 0 | External 12 MHz crystal clock (HXT) 1 | Internal RC 12 MHz oscillator clock (HIRC) The value of CFOSC will be load to HCLK_S(CLKSEL0[2]) in system register after any reset occurs.
[25:21]	Reserved	Reserved
[20:19]	CBORST	**Brown-out Reset Enable Selection** CBORST[1:0] | Brown-out Reset Selection 00 | BOD17 reset enable 01 | BOD20 reset enable 10 | BOD25 reset enable 11 | Disable all BOD functions
[18:16]	Reserved	Reserved
[15:13]	CHXT_GAIN	**CONFIG HXT Gain** HXT_GAIN(PWRCTL[11:10]) is reloaded from the setting of CHXT_GAIN after chip is reset. 000 = reserved 001 = Set HXT gain for higher than 16MHz crystal 010 = Set HXT gain for crystal from 12MHz to 16MHz 011 = Set HXT gain for crystal from 8MHz to 12MHz 100 = Set HXT gain for lower than 8MHz crystal Others = Set HXT gain for crystal from 12MHz to 16MHz

[14:8]	**Reserved**	Reserved
[7:6]	**CBS**	**CONFIG Boot Selection** 00 = LDROM with IAP function. 01 = LDROM without IAP function. 10 = APROM with IAP function. 11 = APROM without IAP function.
[5:2]	**Reserved**	**Reserved**
[1]	**LOCK**	**Security Lock** 0 = Flash data is locked 1 = Flash data is not locked When flash data is locked, only Device ID, CONFIG0 and CONFIG1 can be read by writer and ICP through serial debug interface. Other data is locked as 0xFFFFFFFF. ISP can read data anywhere regardless of LOCK bit value.
[0]	**DFEN**	**Data Flash Enable** 0 = Data flash Enabled. 1 = Data flash Disabled. This bit is valid when CBS[1:0] = 11 or 10.

CONFIG1 (Address = 0x0030_0004)

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				DFBADR			
15	14	13	12	11	10	9	8
DFBADR							
7	6	5	4	3	2	1	0
DFBADR							

CONFIG1	Address = 0x0030_0004	
Bits	**Description**	
[31:20]	Reserved	**Reserved** **It is mandatory to program 0x00 to these Reserved bits**
[19:0]	DFBADR	**Data Flash Base Address** The data flash base address is defined by user. Since on chip flash erase unit is 512 bytes, it is mandatory to keep bit 8-0 as 0.

6.6.4.8 In System Program (ISP)

The application program memory and Data Flash supports both hardware programming mode and In System Programming (ISP) mode. Hardware programming mode uses gang-writers to reduce programming costs and time to market while the products enter into the mass production state. However, if the product is just under development or the end product needs firmware updating in the hand of an end user, the hardware programming mode will make repeated programming difficult and inconvenient. ISP method makes it easy and possible. This chip supports ISP mode allowing a device to be reprogrammed under software control. Furthermore, the capability to update the application firmware makes wide range of applications possible.

ISP is performed without removing the microcontroller from the system. Various interfaces enable LDROM firmware to get new program code easily. The most common method to perform ISP is via UART along with the firmware in LDROM. General speaking, PC transfers the new APROM code through serial port. Then LDROM firmware receives it and re-programs into APROM through ISP commands. Nuvoton provides ISP firmware and PC application program for this chip. It makes users quite easy to perform ISP through Nuvoton ISP tool.

ISP Procedure

This chip supports booting from APROM or LDROM initially defined by user configuration bit (CBS). If user wants to update application program in APROM, user can write 1 to BS(ISPCON[1]) and starts software reset to make chip boot from LDROM. The first step to start ISP function is to write 1 t0 ISPEN(ISPCON[0]). Software is required to write RegLockAddr register in Global Control Register (GCR, 0x5000_0100) with 0x59, 0x16 and 0x88 before writing ISPCON register. This procedure is used to protect flash memory from destroying owning to unintended write during power on/off duration.

Several error conditions are checked after software writes ISPGO(ISPTRG[0]) bit. If error condition occurs, ISP operation is not started and ISP fail flag will be set instead ISPFF(ISPSTA[6]) flag is cleared by Software but it will not be overwritten in next ISP operation. The next ISP procedure can be started even ISPFF(ISPSTA[6]) bit keeps at 1. It is recommended that software to check ISPFF(ISPSTA[6]) and clear it after each ISP operation if it is set to 1.**.**

When ISPGO(ISPTRG[0]) is set, CPU will wait for ISP operation finish, during this period; peripheral still keeps working as usual. If any interrupt request occur, CPU will not service it till ISP operation finish**.**

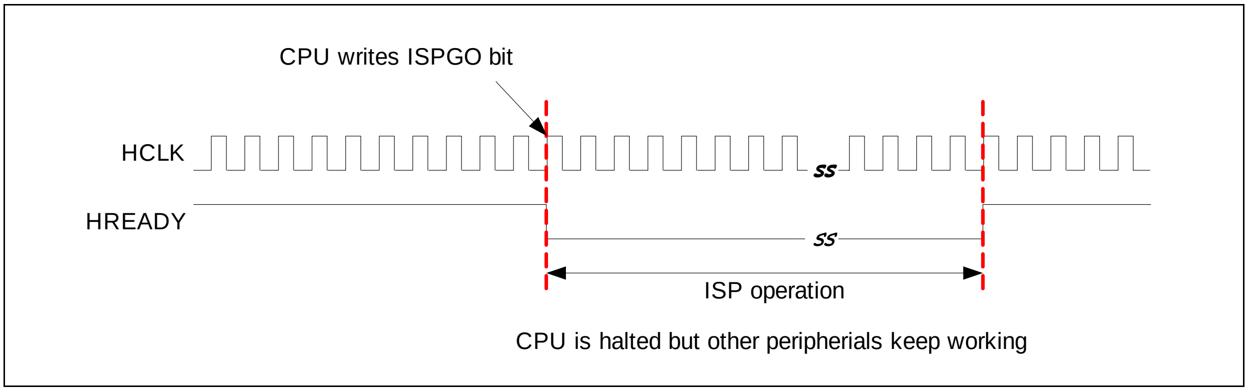


Figure 6‑20 CPU Halt during ISP Operation

Note that this chip allows user to update CONFIG value by ISP, but for application program code security issue, software is required to erase APROM by page erase before erase CONFIG. Otherwise, erase CONFIG will not be allowed.

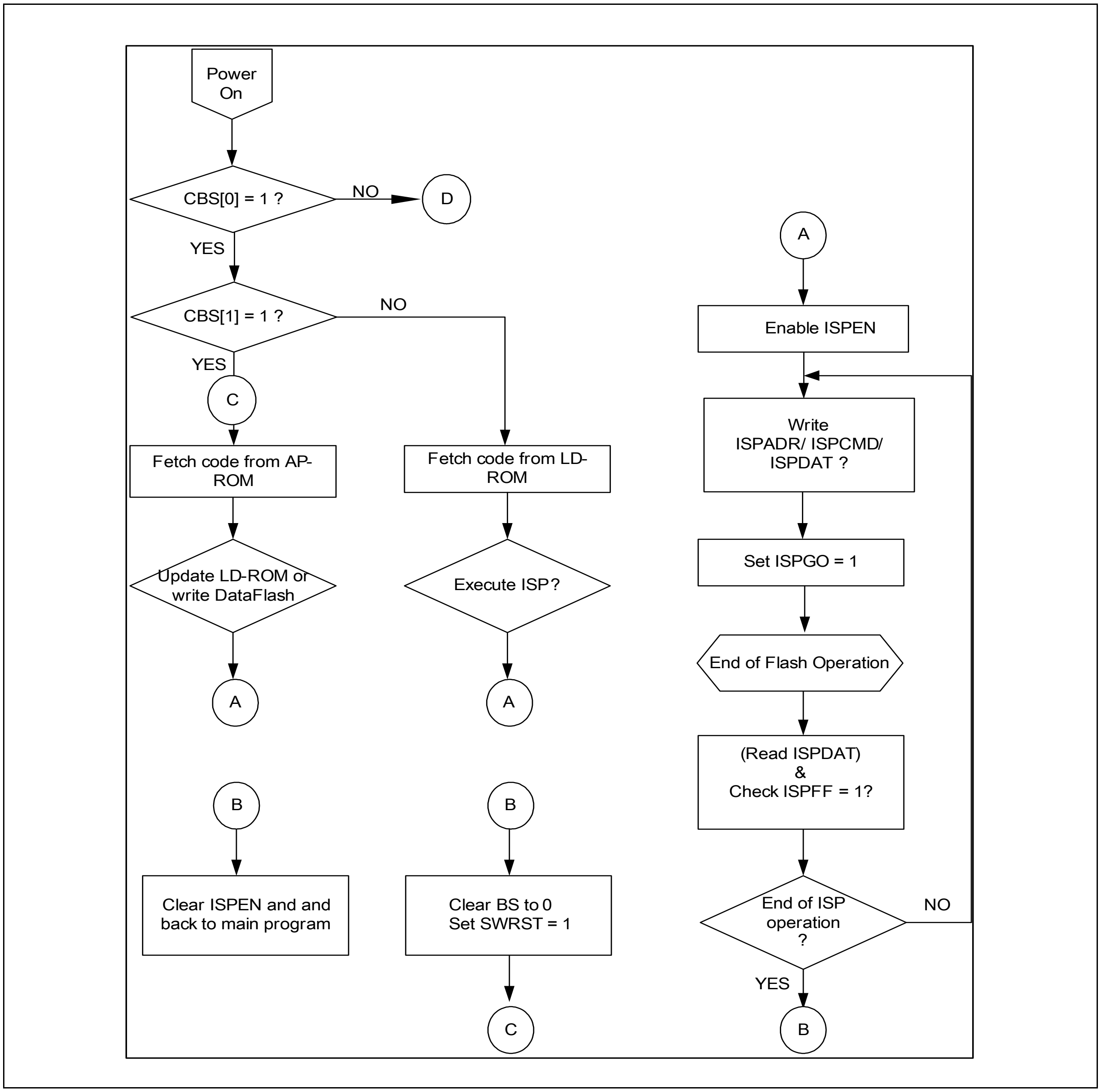


Figure 6-21 ISP Operation Flow

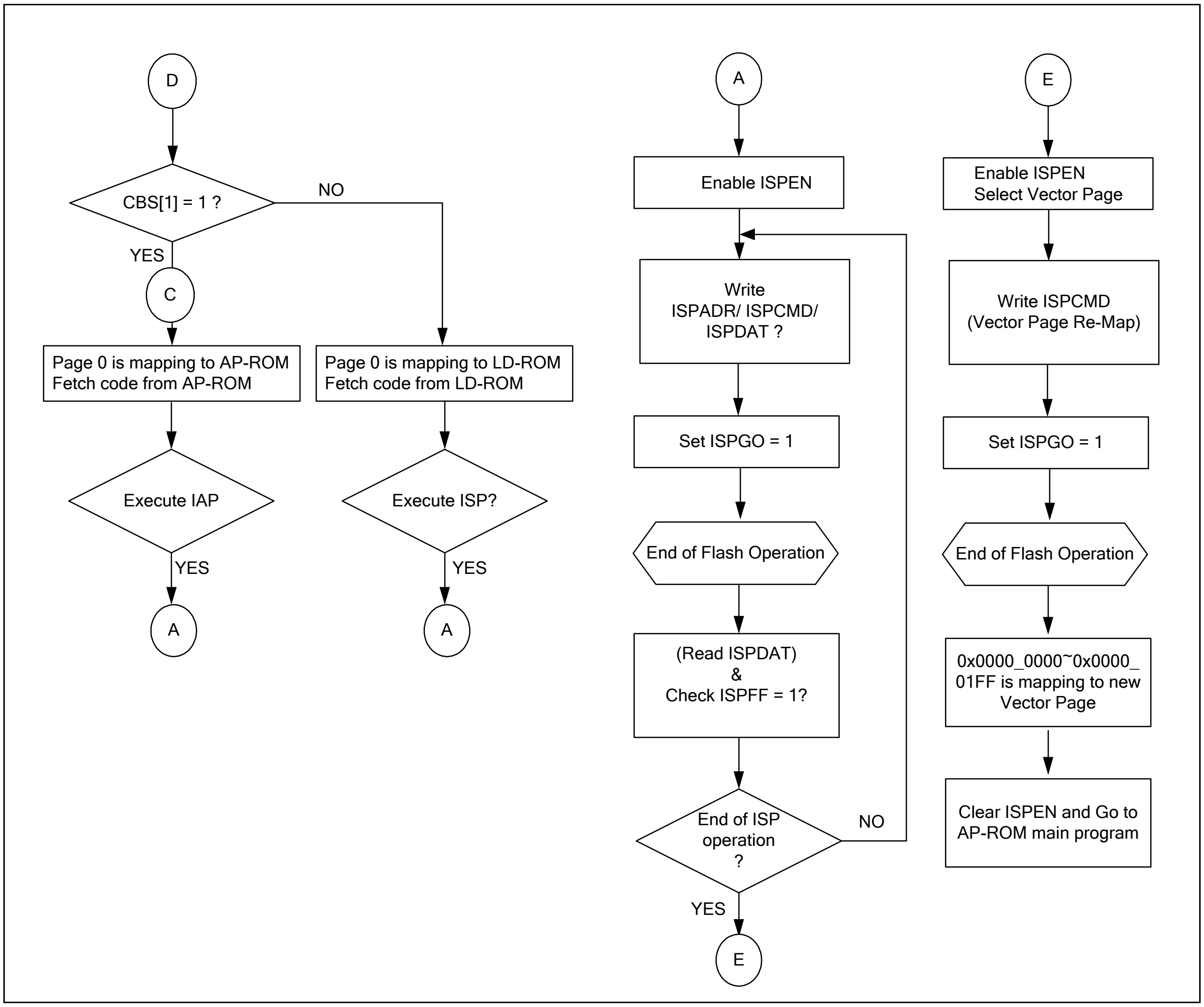


Figure 6-38 ISP Operation Flow (Continued)

ISP Mode	ISPCMD			ISPADR			ISPDAT
	FOEN	FCEN	FCTRL[3:0]	A21	A20	A[19:0]	D[31:0]
Standby	1	1	X	x	x	x	x
Read Company ID	0	0	1011	x	x	x	Data out D[31:0] = 0x0000_00DA
Read Device ID	0	0	1100	x	x	Address in A[19:0] = 0x00000	Data out D[31:0]= Device ID
Read Unique ID	0	0	0100	x	x	Address in A[19:0] = 0x00000 0x00004 0x00008	Data out D[31:0]= Unique ID
*Read Unique Customer ID	0	0	0100	x	x	Address in A[19:0] = 0x00010 0x00014 0x00018 0x0001C	Data out D[31:0]= Unique Customer ID
Vector Page Re-Map	1	0	1110	0	A20	Address in A[19:0]	x
FLASH Page Erase	1	0	0010	0	A20	Address in A[19:0]	x
FLASH Program	1	0	0001	0	A20	Address in A[19:0]	Data in D[31:0]
FLASH Read	0	0	0000	0	A20	Address in A[19:0]	Data out D[31:0]
CONFIG Page Erase	1	0	0010	1	1	Address in A[19:0]	x
CONFIG Program	1	0	0001	1	1	Address in A[19:0]	Data in D[31:0]
CONFIG Read	0	0	0000	1	1	Address in A[19:0]	Data out D[31:0]

Table 6-12 ISP Operation Command

* The default value of “Unique Customer ID” is 0xFFFF which is from address 0x00010 to 0x0001C. “Unique Customer ID” only can be configured by Nuvoton, please contact Nuvoton or agent to deal with specific customer ID.

6.6.5 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
FMC Base Address: **FMC_BA = 0x5000_C000**				
ISPCON	FMC_BA+0x00	R/W	ISP Control Register	0x0000_0000
ISPADR	FMC_BA+0x04	R/W	ISP Address Register	0x0000_0000
ISPDAT	FMC_BA+0x08	R/W	ISP Data Register	0x0000_0000
ISPCMD	FMC_BA+0x0C	R/W	ISP Command Register	0x0000_0000
ISPTRG	FMC_BA+0x10	R/W	ISP Trigger Register	0x0000_0000
DFBADR	FMC_BA+0x14	R	Data Flash Base Address Regrister	0x0001_F000
ISPSTA	FMC_BA+0x40	R/W	ISP Status Register	0x0000_0000

6.6.6 Register Description

ISP Control Register (ISPCON)

Register	Offset	R/W	Description	Reset Value
ISPCON	FMC_BA+0x00	R/W	ISP Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	ISPFF	LDUEN	CFGUEN	APUEN	Reserved	BS	ISPEN

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**ISPFF**	**ISP Fail Flag (Write Protect)** **This bit is set by hardware when a triggered ISP meets any of the following conditions:** (1) APROM writes to itself if APUEN is set to 0 or CBS[0]=1. (2) LDROM writes to itself if LDUEN is set to 0 or CBS[0]=1. (3) User Configuration is erased/programmed when CFGUEN is 0. (4) Destination address is illegal, such as over an available range. **Note:** Write 1 to clear this bit to 0.
[5]	**LDUEN**	**LDROM Update Enable Control (Write Protect)** 0 = LDROM cannot be updated. 1 = LDROM can be updated.
[4]	**CFGUEN**	**Enable Config-bits Update by ISP (Write Protect)** 0 = ISP update User Configuration Disabled. 1 = ISP update User Configuration Enabled.
[3]	**APUEN**	**APROM Update Enable Control (Write Protect)** 0 = APROM cannot be updated. 1 = APROM can be updated.
[2]	**Reserved**	Reserved.
[1]	**BS**	**Boot Select (Write Protect)** Set/clear this bit to select next booting from LDROM/APROM, respectively. This bit also functions as chip booting status flag, which can be used to check where chip booted from. This bit is initiated with the inversed value of CBS in Config0 after power-on reset; It keeps the same value at other reset. 0 = Boot from APROM. 1 = Boot from LDROM.

[0]	**ISPEN**	**ISP Enable Control (Write Protect)** **ISP function enable bit. Set this bit to enable ISP function.** 0 = ISP function Disabled. 1 = ISP function Enabled.

ISP Address (ISPADR)

Register	Offset	R/W	Description	Reset Value
ISPADR	FMC_BA+0x04	R/W	ISP Address Register	0x0000_0000

31	30	29	28	27	26	25	24
ISPADR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ISPADR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ISPADR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ISPADR							

Bits	Description	
[31:0]	**ISPADR**	**ISP Address** This chip supports word program only. ISPADR[1:0] must be kept 00b for ISP operation, and ISPADR[8:0] must be kept all 0 for Vector Page Re-map Command

ISPDAT (ISP Data Register)

Register	Offset	R/W	Description	Reset Value
ISPDAT	FMC_BA+0x08	R/W	ISP Data Register	0x0000_0000

31	30	29	28	27	26	25	24
ISPDAT							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ISPDAT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ISPDAT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ISPDAT							

Bits	Description	
[31:0]	**ISPDAT**	**ISP Data** Write data to this register before ISP program operation Read data from this register after ISP read operation

ISP Command (ISPCMD)

Register	Offset	R/W	Description	Reset Value
ISPCMD	FMC_BA+0x0C	R/W	ISP Command Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		FOEN	FCEN	FCTRL			

Bits	Description	
[31:6]	Reserved	Reserved.
[5]	FOEN	**ISP Command** The ISP command table is shown as follows.
[4]	FCEN	**ISP Command** The ISP command table is shown as follows.
[3:0]	FCTRL	**ISP Command** The ISP command table is shown as follows.

Operation Mode	FOEN	FCEN	FCTRL[3:0]			
Read	0	0	0	0	0	0
Vector Page Re-Map	1	0	1	1	1	0
Program	1	0	0	0	0	1
Page Erase	1	0	0	0	1	0
Read CID	0	0	1	0	1	1
Read DID	0	0	1	1	0	0
Read UID	0	0	0	1	0	0

ISP Trigger Control Register (ISPTRG)

Register	Offset	R/W	Description	Reset Value
ISPTRG	FMC_BA+0x10	R/W	ISP Trigger Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**ISPGO**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**ISPGO**	**ISP Start Trigger** Write 1 to start ISP operation and this bit will be cleared to 0 by hardware automatically when ISP operation is finished. 0 = ISP operation is finished. 1 = ISP is progressing.

Data Flash Base Address Register (DFBADR)

Register	Offset	R/W	Description	Reset Value
DFBADR	FMC_BA+0x14	R	Data Flash Base Address Register	0x0001_F000

31	30	29	28	27	26	25	24
DFBADR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DFBADR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DFBADR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DFBADR							

Bits	Description	
[31:0]	**DFBADR**	**Data Flash Base Address** This register indicates data flash start address. It is a read only register. The data flash start address is defined by user. Since on chip flash erase unit is 512 bytes, it is mandatory to keep bit 8-0 as 0.

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

ISP Status Register (ISPSTA)

Register	Offset	R/W	Description	Reset Value
ISPSTA	FMC_BA+0x40	R/W	ISP Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved			VECMAP				
15	14	13	12	11	10	9	8
VECMAP							
7	6	5	4	3	2	1	0
Reserved	ISPFF	PGFF	Reserved		CBS		ISPBUSY

Bits	Description	
[31:21]	Reserved	Reserved.
[20:9]	VECMAP	**Vector Page Mapping Address (Read Only)** The current system memory space 0x0000_0000~0x0000_01FF is mapped to flash memory page with base address (VECMAP[11:0] << 9).
[8:7]	Reserved	Reserved.
[6]	ISPFF	**ISP Fail Flag** (1) APROM writes to itself if APUEN is set to 0 or CBS[0]=1. (2) LDROM writes to itself if LDUEN is set to 0 or CBS[0]=1. (3) User Configuration is erased/programmed when CFGUEN is 0. (4) Destination address is illegal, such as over an available range. **Note:** Write 1 to clear this bit to 0.
[5]	PGFF	**Auto Flash Program Verified Fail Flag** This chip will perform flash verification automatically at the end of ISP PROGRAM operation, and set 1 to this bit when flash data is not matched with programming. This bit is clear to 0 by “ERASE” command.
[5:3]	Reserved	Reserved.
[2:1]	CBS	**Config Boot Selection Status (Read Only)** This filed is a mirror of CBS in CONFIG0.
[0]	ISPBUSY	**ISP Busy (Read Only)** 0 = ISP operation is finished. 1 = ISP operation is busy.

6.7 General Purpose I/O Controller

6.7.1 Overview

The NuMicro® Nano112 series have up to 80 General Purpose I/O pins to be shared with other function pins depending on the chip configuration. These 80 pins are arranged in 6 ports named with GPIOA, GPIOB, GPIOC, GPIOD, GPIOE and GPIOF. Each one of the 80 pins is independent and has the corresponding register bits to control the pin mode function and data.

The I/O type of each of I/O pins can be independently software configured as input, output, and open-drain mode. Each I/O pin has a very weak individual pull-up resistor which is about 110 KΩ~300 KΩ for V_{DD} from 1.8 V to 3.6 V.

6.7.2 Features

- Three I/O modes:
  - Schmitt trigger Input-only with high impendence
  - Push-pull output
  - Open-drain output
- I/O pin configured as interrupt source with edge/level setting
- Enabling the pin interrupt function will also enable the pin wake-up function

6.7.3 Basic Configuration

The GPIO pin functions are configured in PA_L_MFP, PA_H_MFP, PB_L_MFP, PB_H_MFP, PC_L_MFP, PC_H_MFP, PD_L_MFP, PD_H_MFP, PE_L_MFP, PE_H_MFP, PF_L_MFP registers.

6.7.4 Block Diagram

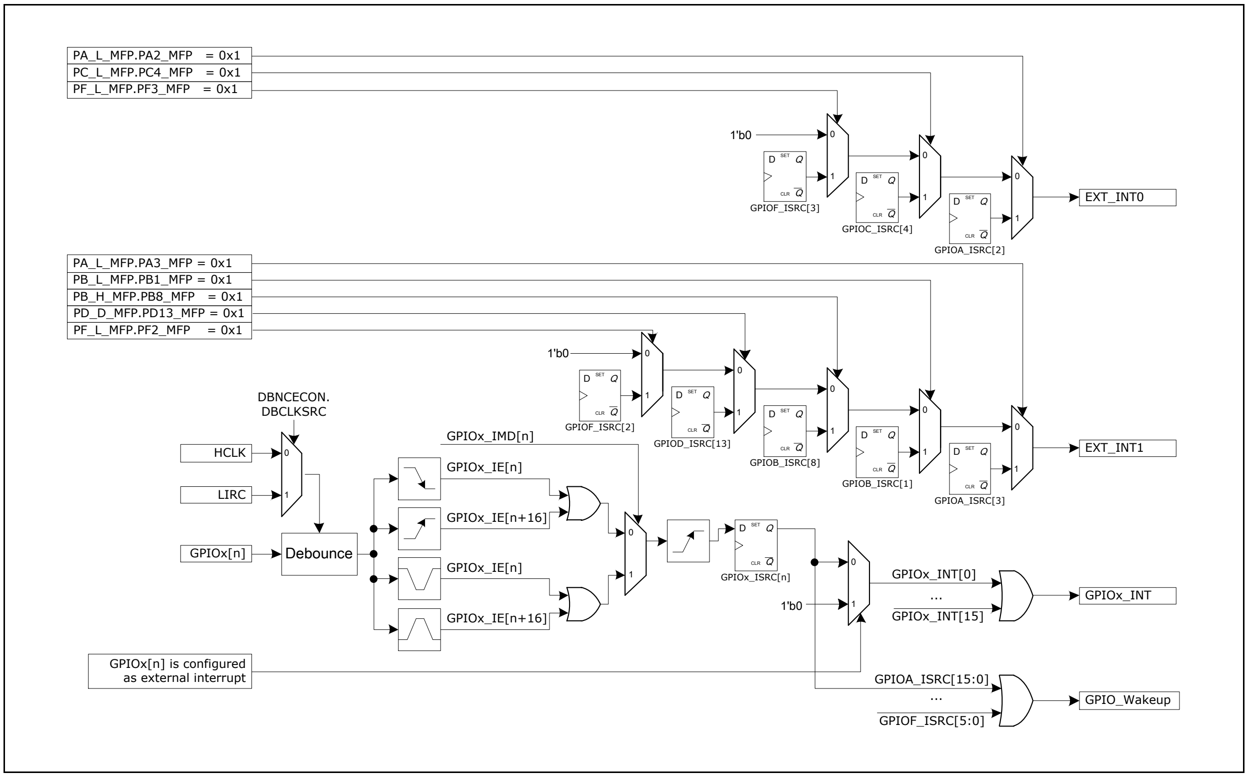


Figure 6-22 GPIO Block Diagram

6.7.5 Functional Description

6.7.5.1 Input Mode Explanation

Set GPIOx_PMD (PMDn [1:0]) to 00 the GPIOx port [n] pin is in Input mode and the I/O pin is in tri-state (high impedance) without output drive capability. The GPIOx_PIN value reflects the status of the corresponding port pins.

6.7.5.2 Output Mode Explanation

Set GPIOx_PMD (PMDn [1:0]) to 01 the GPIOx port [n] pin is in Output mode and the I/O pin supports digital output function with source/sink current capability. The bit value in the corresponding bit [n] of GPIO_DOUT is driven on the pin.

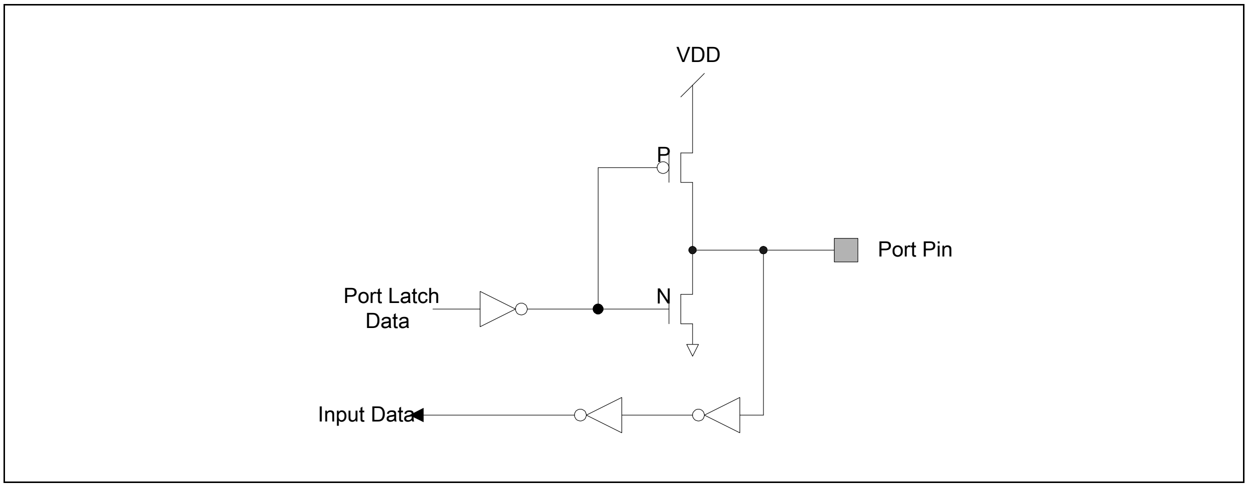


Figure 6‑23 Push-Pull Output

6.7.5.3 Open-Drain Mode Explanation

Set GPIOx_PMD (PMDn [1:0]) to 10 the GPIOx port [n] pin is in Open-Drain mode and the I/O pin supports digital output function but only with sink current capability, an additional pull-up resister is needed for driving high state. If the bit value in the corresponding bit [n] of GPIOx_DOUT is "0", the pin drive a "low" output on the pin. If the bit value in the corresponding bit [n] of GPIOx_DOUT is "1", the pin output drives high that is controlled by the internal pull-up resistor or the external pull high resistor.

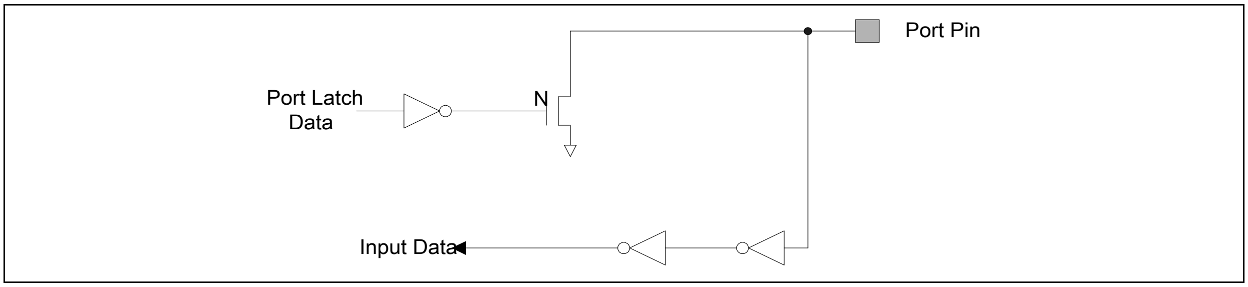


Figure 6‑24 Open-Drain Output

6.7.6 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
GP Base Address: **GP_BA =0x5000_4000**				
GPIOA_PMD	GP_BA+0x000	R/W	GPIO Port A Pin I/O Mode Control Register	0x0000_0000
GPIOA_OFFD	GP_BA+0x004	R/W	GPIO Port A Pin OFF Digital Enable Register	0x0000_0000
GPIOA_DOUT	GP_BA+0x008	R/W	GPIO Port A Data Output Value Register	0x0000_FFFF
GPIOA_DMASK	GP_BA+0x00C	R/W	GPIO Port A Data Output Write Mask Register	0x0000_0000
GPIOA_PIN	GP_BA+0x010	R	GPIO Port A Pin Value Register	0x0000_XXXX
GPIOA_DBEN	GP_BA+0x014	R/W	GPIO Port A De-bounce Enable Register	0x0000_0000
GPIOA_IMD	GP_BA+0x018	R/W	GPIO Port A Interrupt Mode Control Register	0x0000_0000
GPIOA_IER	GP_BA+0x01C	R/W	GPIO Port A Interrupt Enable Register	0x0000_0000
GPIOA_ISRC	GP_BA+0x020	R/W	GPIO Port A Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOA_PUEN	GP_BA+0x024	R/W	GPIO Port A Pull-Up Enable Register	0x0000_0000
GPIOB_PMD	GP_BA+0x040	R/W	GPIO Port B Pin I/O Mode Control Register	0x0000_0000
GPIOB_OFFD	GP_BA+0x044	R/W	GPIO Port B Pin OFF Digital Enable Register	0x0000_0000
GPIOB_DOUT	GP_BA+0x048	R/W	GPIO Port B Data Output Value Register	0x0000_FFFF
GPIOB_DMASK	GP_BA+0x04C	R/W	GPIO Port B Data Output Write Mask Register	0x0000_0000
GPIOB_PIN	GP_BA+0x050	R	GPIO Port B Pin Value Register	0x0000_XXXX
GPIOB_DBEN	GP_BA+0x054	R/W	GPIO Port B De-bounce Enable Register	0x0000_0000
GPIOB_IMD	GP_BA+0x058	R/W	GPIO Port B Interrupt Mode Control Register	0x0000_0000
GPIOB_IER	GP_BA+0x05C	R/W	GPIO Port B Interrupt Enable Register	0x0000_0000
GPIOB_ISRC	GP_BA+0x060	R/W	GPIO Port B Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOB_PUEN	GP_BA+0x064	R/W	GPIO Port B Pull-Up Enable Register	0x0000_0000
GPIOC_PMD	GP_BA+0x080	R/W	GPIO Port C Pin I/O Mode Control Register	0x0000_0000
GPIOC_OFFD	GP_BA+0x084	R/W	GPIO Port C Pin OFF Digital Enable Register	0x0000_0000
GPIOC_DOUT	GP_BA+0x088	R/W	GPIO Port C Data Output Value Register	0x0000_FFFF
GPIOC_DMASK	GP_BA+0x08C	R/W	GPIO Port C Data Output Write Mask Register	0x0000_0000
GPIOC_PIN	GP_BA+0x090	R	GPIO Port C Pin Value Register	0x0000_XXXX
GPIOC_DBEN	GP_BA+0x094	R/W	GPIO Port C De-bounce Enable Register	0x0000_0000

GPIOC_IMD	GP_BA+0x098	R/W	GPIO Port C Interrupt Mode Control Register	0x0000_0000
GPIOC_IER	GP_BA+0x09C	R/W	GPIO Port C Interrupt Enable Register	0x0000_0000
GPIOC_ISRC	GP_BA+0x0A0	R/W	GPIO Port C Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOC_PUEN	GP_BA+0x0A4	R/W	GPIO Port C Pull-Up Enable Register	0x0000_0000
GPIOD_PMD	GP_BA+0x0C0	R/W	GPIO Port D Pin I/O Mode Control Register	0x0000_0000
GPIOD_OFFD	GP_BA+0x0C4	R/W	GPIO Port D Pin OFF Digital Enable Register	0x0000_0000
GPIOD_DOUT	GP_BA+0x0C8	R/W	GPIO Port D Data Output Value Register	0x0000_FFFF
GPIOD_DMASK	GP_BA+0x0CC	R/W	GPIO Port D Data Output Write Mask Register	0x0000_0000
GPIOD_PIN	GP_BA+0x0D0	R	GPIO Port D Pin Value Register	0x0000_XXXX
GPIOD_DBEN	GP_BA+0x0D4	R/W	GPIO Port D De-bounce Enable Register	0x0000_0000
GPIOD_IMD	GP_BA+0x0D8	R/W	GPIO Port D Interrupt Mode Control Register	0x0000_0000
GPIOD_IER	GP_BA+0x0DC	R/W	GPIO Port D Interrupt Enable Register	0x0000_0000
GPIOD_ISRC	GP_BA+0x0E0	R/W	GPIO Port D Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOD_PUEN	GP_BA+0x0E4	R/W	GPIO Port D Pull-Up Enable Register	0x0000_0000
GPIOE_PMD	GP_BA+0x100	R/W	GPIO Port E Pin I/O Mode Control Register	0x0000_0000
GPIOE_OFFD	GP_BA+0x104	R/W	GPIO Port E Pin OFF Digital Enable Register	0x0000_0000
GPIOE_DOUT	GP_BA+0x108	R/W	GPIO Port E Data Output Value Register	0x0000_FFFF
GPIOE_DMASK	GP_BA+0x10C	R/W	GPIO Port E Data Output Write Mask Register	0x0000_0000
GPIOE_PIN	GP_BA+0x110	R	GPIO Port E Pin Value Register	0x0000_XXXX
GPIOE_DBEN	GP_BA+0x114	R/W	GPIO Port E De-bounce Enable Register	0x0000_0000
GPIOE_IMD	GP_BA+0x118	R/W	GPIO Port E Interrupt Mode Control Register	0x0000_0000
GPIOE_IER	GP_BA+0x11C	R/W	GPIO Port E Interrupt Enable Register	0x0000_0000
GPIOE_ISRC	GP_BA+0x120	R/W	GPIO Port E Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOE_PUEN	GP_BA+0x124	R/W	GPIO Port E Pull-Up Enable Register	0x0000_0000
GPIOF_PMD	GP_BA+0x140	R/W	GPIO Port F Pin I/O Mode Control Register	0x0000_0000
GPIOF_OFFD	GP_BA+0x144	R/W	GPIO Port F Pin OFF Digital Enable Register	0x0000_0000
GPIOF_DOUT	GP_BA+0x148	R/W	GPIO Port F Data Output Value Register	0x0000_003F
GPIOF_DMASK	GP_BA+0x14C	R/W	GPIO Port F Data Output Write Mask Register	0x0000_0000
GPIOF_PIN	GP_BA+0x150	R	GPIO Port F Pin Value Register	0x0000_00XX
GPIOF_DBEN	GP_BA+0x154	R/W	GPIO Port F De-bounce Enable Register	0x0000_0000
GPIOF_IMD	GP_BA+0x158	R/W	GPIO Port F Interrupt Mode Control Register	0x0000_0000

GPIOF_IER	GP_BA+0x15C	R/W	GPIO Port F Interrupt Enable Register	0x0000_0000
GPIOF_ISRC	GP_BA+0x160	R/W	GPIO Port F Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOF_PUEN	GP_BA+0x164	R/W	GPIO Port F Pull-Up Enable Register	0x0000_0000
DBNCECON	GP_BA+0x180	R/W	De-bounce Cycle Control Register	0x0000_0000
GPIOA0	GP_BA+0x200	R/W	GPIO Port A Bit 0 Data Register	0x0000_000X
GPIOA1	GP_BA+0x204	R/W	GPIO Port A Bit 1 Data Register	0x0000_000X
GPIOA2	GP_BA+0x208	R/W	GPIO Port A Bit 2 Data Register	0x0000_000X
GPIOA3	GP_BA+0x20C	R/W	GPIO Port A Bit 3 Data Register	0x0000_000X
GPIOA4	GP_BA+0x210	R/W	GPIO Port A Bit 4 Data Register	0x0000_000X
GPIOA5	GP_BA+0x214	R/W	GPIO Port A Bit 5 Data Register	0x0000_000X
GPIOA6	GP_BA+0x218	R/W	GPIO Port A Bit 6 Data Register	0x0000_000X
GPIOA7	GP_BA+0x21C	R/W	GPIO Port A Bit 7 Data Register	0x0000_000X
GPIOA8	GP_BA+0x220	R/W	GPIO Port A Bit 8 Data Register	0x0000_000X
GPIOA9	GP_BA+0x224	R/W	GPIO Port A Bit 9 Data Register	0x0000_000X
GPIOA10	GP_BA+0x228	R/W	GPIO Port A Bit 10 Data Register	0x0000_000X
GPIOA11	GP_BA+0x22C	R/W	GPIO Port A Bit 11 Data Register	0x0000_000X
GPIOA12	GP_BA+0x230	R/W	GPIO Port A Bit 12 Data Register	0x0000_000X
GPIOA13	GP_BA+0x234	R/W	GPIO Port A Bit 13 Data Register	0x0000_000X
GPIOA14	GP_BA+0x238	R/W	GPIO Port A Bit 14 Data Register	0x0000_000X
GPIOA15	GP_BA+0x23C	R/W	GPIO Port A Bit 15 Data Register	0x0000_000X
GPIOB0	GP_BA+0x240	R/W	GPIO Port B Bit 0 Data Register	0x0000_000X
GPIOB1	GP_BA+0x244	R/W	GPIO Port B Bit 1 Data Register	0x0000_000X
GPIOB2	GP_BA+0x248	R/W	GPIO Port B Bit 2 Data Register	0x0000_000X
GPIOB3	GP_BA+0x24C	R/W	GPIO Port B Bit 3 Data Register	0x0000_000X
GPIOB4	GP_BA+0x250	R/W	GPIO Port B Bit 4 Data Register	0x0000_000X
GPIOB5	GP_BA+0x254	R/W	GPIO Port B Bit 5 Data Register	0x0000_000X
GPIOB6	GP_BA+0x258	R/W	GPIO Port B Bit 6 Data Register	0x0000_000X
GPIOB7	GP_BA+0x25C	R/W	GPIO Port B Bit 7 Data Register	0x0000_000X
GPIOB8	GP_BA+0x260	R/W	GPIO Port B Bit 8 Data Register	0x0000_000X
GPIOB9	GP_BA+0x264	R/W	GPIO Port B Bit 9 Data Register	0x0000_000X
GPIOB10	GP_BA+0x268	R/W	GPIO Port B Bit 10 Data Register	0x0000_000X

GPIOB11	GP_BA+0x26C	R/W	GPIO Port B Bit 11 Data Register	0x0000_000X
GPIOB12	GP_BA+0x270	R/W	GPIO Port B Bit 12 Data Register	0x0000_000X
GPIOB13	GP_BA+0x274	R/W	GPIO Port B Bit 13 Data Register	0x0000_000X
GPIOB14	GP_BA+0x278	R/W	GPIO Port B Bit 14 Data Register	0x0000_000X
GPIOB15	GP_BA+0x27C	R/W	GPIO Port B Bit 15 Data Register	0x0000_000X
GPIOC0	GP_BA+0x280	R/W	GPIO Port C Bit 0 Data Register	0x0000_000X
GPIOC1	GP_BA+0x284	R/W	GPIO Port C Bit 1 Data Register	0x0000_000X
GPIOC2	GP_BA+0x288	R/W	GPIO Port C Bit 2 Data Register	0x0000_000X
GPIOC3	GP_BA+0x28C	R/W	GPIO Port C Bit 3 Data Register	0x0000_000X
GPIOC4	GP_BA+0x290	R/W	GPIO Port C Bit 4 Data Register	0x0000_000X
GPIOC5	GP_BA+0x294	R/W	GPIO Port C Bit 5 Data Register	0x0000_000X
GPIOC6	GP_BA+0x298	R/W	GPIO Port C Bit 6 Data Register	0x0000_000X
GPIOC7	GP_BA+0x29C	R/W	GPIO Port C Bit 7 Data Register	0x0000_000X
GPIOC8	GP_BA+0x2A0	R/W	GPIO Port C Bit 8 Data Register	0x0000_000X
GPIOC9	GP_BA+0x2A4	R/W	GPIO Port C Bit 9 Data Register	0x0000_000X
GPIOC10	GP_BA+0x2A8	R/W	GPIO Port C Bit 10 Data Register	0x0000_000X
GPIOC11	GP_BA+0x2AC	R/W	GPIO Port C Bit 11 Data Register	0x0000_000X
GPIOC12	GP_BA+0x2B0	R/W	GPIO Port C Bit 12 Data Register	0x0000_000X
GPIOC13	GP_BA+0x2B4	R/W	GPIO Port C Bit 13 Data Register	0x0000_000X
GPIOC14	GP_BA+0x2B8	R/W	GPIO Port C Bit 14 Data Register	0x0000_000X
GPIOC15	GP_BA+0x2BC	R/W	GPIO Port C Bit 15 Data Register	0x0000_000X
GPIOD0	GP_BA+0x2C0	R/W	GPIO Port D Bit 0 Data Register	0x0000_000X
GPIOD1	GP_BA+0x2C4	R/W	GPIO Port D Bit 1 Data Register	0x0000_000X
GPIOD2	GP_BA+0x2C8	R/W	GPIO Port D Bit 2 Data Register	0x0000_000X
GPIOD3	GP_BA+0x2CC	R/W	GPIO Port D Bit 3 Data Register	0x0000_000X
GPIOD4	GP_BA+0x2D0	R/W	GPIO Port D Bit 4 Data Register	0x0000_000X
GPIOD5	GP_BA+0x2D4	R/W	GPIO Port D Bit 5 Data Register	0x0000_000X
GPIOD6	GP_BA+0x2D8	R/W	GPIO Port D Bit 6 Data Register	0x0000_000X
GPIOD7	GP_BA+0x2DC	R/W	GPIO Port D Bit 7 Data Register	0x0000_000X
GPIOD8	GP_BA+0x2E0	R/W	GPIO Port D Bit 8 Data Register	0x0000_000X
GPIOD9	GP_BA+0x2E4	R/W	GPIO Port D Bit 9 Data Register	0x0000_000X

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

GPIOD10	GP_BA+0x2E8	R/W	GPIO Port D Bit 10 Data Register	0x0000_000X
GPIOD11	GP_BA+0x2EC	R/W	GPIO Port D Bit 11 Data Register	0x0000_000X
GPIOD12	GP_BA+0x2F0	R/W	GPIO Port D Bit 12 Data Register	0x0000_000X
GPIOD13	GP_BA+0x2F4	R/W	GPIO Port D Bit 13 Data Register	0x0000_000X
GPIOD14	GP_BA+0x2F8	R/W	GPIO Port D Bit 14 Data Register	0x0000_000X
GPIOD15	GP_BA+0x2FC	R/W	GPIO Port D Bit 15 Data Register	0x0000_000X
GPIOE0	GP_BA+0x300	R/W	GPIO Port E Bit 0 Data Register	0x0000_000X
GPIOE1	GP_BA+0x304	R/W	GPIO Port E Bit 1 Data Register	0x0000_000X
GPIOE2	GP_BA+0x308	R/W	GPIO Port E Bit 2 Data Register	0x0000_000X
GPIOE3	GP_BA+0x30C	R/W	GPIO Port E Bit 3 Data Register	0x0000_000X
GPIOE4	GP_BA+0x310	R/W	GPIO Port E Bit 4 Data Register	0x0000_000X
GPIOE5	GP_BA+0x314	R/W	GPIO Port E Bit 5 Data Register	0x0000_000X
GPIOE6	GP_BA+0x318	R/W	GPIO Port E Bit 6 Data Register	0x0000_000X
GPIOE7	GP_BA+0x31C	R/W	GPIO Port E Bit 7 Data Register	0x0000_000X
GPIOE8	GP_BA+0x320	R/W	GPIO Port E Bit 8 Data Register	0x0000_000X
GPIOE9	GP_BA+0x324	R/W	GPIO Port E Bit 9 Data Register	0x0000_000X
GPIOF0	GP_BA+0x340	R/W	GPIO Port F Bit 0 Data Register	0x0000_000X
GPIOF1	GP_BA+0x344	R/W	GPIO Port F Bit 1 Data Register	0x0000_000X
GPIOF2	GP_BA+0x348	R/W	GPIO Port F Bit 2 Data Register	0x0000_000X
GPIOF3	GP_BA+0x34C	R/W	GPIO Port F Bit 3 Data Register	0x0000_000X
GPIOF4	GP_BA+0x350	R/W	GPIO Port F Bit 4 Data Register	0x0000_000X
GPIOF5	GP_BA+0x354	R/W	GPIO Port F Bit 5 Data Register	0x0000_000X

6.7.7 Register Description

GPIO Port [A/B/C/D/E/F] Pin I/O Mode Control Register (GPIOx_PMD)

Register	Offset	R/W	Description	Reset Value
GPIOA_PMD	GP_BA+0x000	R/W	GPIO Port A Pin I/O Mode Control Register	0x0000_0000
GPIOB_PMD	GP_BA+0x040	R/W	GPIO Port B Pin I/O Mode Control Register	0x0000_0000
GPIOC_PMD	GP_BA+0x080	R/W	GPIO Port C Pin I/O Mode Control Register	0x0000_0000
GPIOD_PMD	GP_BA+0x0C0	R/W	GPIO Port D Pin I/O Mode Control Register	0x0000_0000
GPIOE_PMD	GP_BA+0x100	R/W	GPIO Port E Pin I/O Mode Control Register	0x0000_0000
GPIOF_PMD	GP_BA+0x140	R/W	GPIO Port F Pin I/O Mode Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PMD15		PMD14		PMD13		PMD12	
23	22	21	20	19	18	17	16
PMD11		PMD10		PMD9		PMD8	
15	14	13	12	11	10	9	8
PMD7		PMD6		PMD5		PMD4	
7	6	5	4	3	2	1	0
PMD3		PMD2		PMD1		PMD0	

Bits	Description	
[2n+1:2n] n = 0,1..15	PMDn	**GPIO Port [X] Pin [N] Mode Control** Determine the I/O type of GPIO port [x] pin [n] 00 = GPIO port [x] pin [n] is in INPUT mode. 01 = GPIO port [x] pin [n] is in OUTPUT mode. 10 = GPIO port [x] pin [n] is in Open-Drain mode. 11 = Reserved. **Note:** For GPIOE_PMD, PMD10 ~ PMD15 are reserved. For GPIOF_PMD, PMD6 ~ PMD15 are reserved.

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

GPIO Port [A/B/C/D/E/F] Pin OFF Digital Enable Resistor (GPIOx_OFFD)

Register	Offset	R/W	Description	Reset Value
GPIOA_OFFD	GP_BA+0x004	R/W	GPIO Port A Pin OFF Digital Enable Register	0x0000_0000
GPIOB_OFFD	GP_BA+0x044	R/W	GPIO Port B Pin OFF Digital Enable Register	0x0000_0000
GPIOC_OFFD	GP_BA+0x084	R/W	GPIO Port C Pin OFF Digital Enable Register	0x0000_0000
GPIOD_OFFD	GP_BA+0x0C4	R/W	GPIO Port D Pin OFF Digital Enable Register	0x0000_0000
GPIOE_OFFD	GP_BA+0x104	R/W	GPIO Port E Pin OFF Digital Enable Register	0x0000_0000
GPIOF_OFFD	GP_BA+0x144	R/W	GPIO Port F Pin OFF Digital Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
OFFD							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
OFFD							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[n+16] n = 0,1..15	**OFFD**	**GPIO Port [X] Pin [N] Digital Input Path Disable** Determine if the digital input path of GPIO port [x] pin [n] is disabled. 0 = Digital input path of GPIO port [x] pin [n] Enabled. 1 = Digital input path of GPIO port [x] pin [n] Disabled (tied digital input to low). **Note:** For GPIOE_OFFD, bits [31:26] are reserved. For GPIOF_OFFD, bits [31:22] are reserved.
[15:0]	**Reserved**	Reserved.

GPIO Port [A/B/C/D/E/F] Data Output Value Register (GPIOx_DOUT)

Register	Offset	R/W	Description	Reset Value
GPIOA_DOUT	GP_BA+0x008	R/W	GPIO Port A Data Output Value Register	0x0000_FFFF
GPIOB_DOUT	GP_BA+0x048	R/W	GPIO Port B Data Output Value Register	0x0000_FFFF
GPIOC_DOUT	GP_BA+0x088	R/W	GPIO Port C Data Output Value Register	0x0000_FFFF
GPIOD_DOUT	GP_BA+0x0C8	R/W	GPIO Port D Data Output Value Register	0x0000_FFFF
GPIOE_DOUT	GP_BA+0x108	R/W	GPIO Port E Data Output Value Register	0x0000_FFFF
GPIOF_DOUT	GP_BA+0x148	R/W	GPIO Port F Data Output Value Register	0x0000_003F

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DOUT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DOUT							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[n] n = 0,1..15	**DOUT**	**GPIO Port [X] Pin [N] Output Value** Each of these bits controls the status of a GPIO port [x] pin [n] when the GPI/O pin is configures as output or open-drain mode 0 = GPIO port [x] Pin [n] will drive Low if the corresponding output mode enabling bit is set. 1 = GPIO port [x] Pin [n] will drive High if the corresponding output mode enabling bit is set. **Note:** For GPIOE_DOUT, bits [15:10] are reserved. For GPIOF_DOUT, bits [15:6] are reserved.

GPIO Port [A/B/C/D/E/F] Data Output Write Mask Register (GPIOx_DMASK)

Register	Offset	R/W	Description	Reset Value
GPIOA_DMASK	GP_BA+0x00C	R/W	GPIO Port A Data Output Write Mask Register	0x0000_0000
GPIOB_DMASK	GP_BA+0x04C	R/W	GPIO Port B Data Output Write Mask Register	0x0000_0000
GPIOC_DMASK	GP_BA+0x08C	R/W	GPIO Port C Data Output Write Mask Register	0x0000_0000
GPIOD_DMASK	GP_BA+0x0CC	R/W	GPIO Port D Data Output Write Mask Register	0x0000_0000
GPIOE_DMASK	GP_BA+0x10C	R/W	GPIO Port E Data Output Write Mask Register	0x0000_0000
GPIOF_DMASK	GP_BA+0x14C	R/W	GPIO Port F Data Output Write Mask Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMASK							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMASK							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[n] n = 0,1..15	**DMASK**	**GPIO Port [X] Pin [N] Data Output Write Mask** These bits are used to protect the corresponding register of GPIOx_DOUT bit [n]. When set the DMASK[n] to “1”, the corresponding DOUT[n] bit is protected. The write signal is masked, write data to the protect bit is ignored 0 = The corresponding GPIO_DOUT bit [n] can be updated. 1 = The corresponding GPIO_DOUT bit [n] is protected. **Note:** For GPIOF_DMASK, bits [15:6] are reserved. **Note:** These mask bits only take effect while CPU is doing write operation to register GPIOx_DOUT. If CPU is doing write operation to register GPIO[x][n], these mask bits will not take effect.

GPIO Port [A/B/C/D/E/F] Pin Value Register (GPIOx_PIN)

Register	Offset	R/W	Description	Reset Value
GPIOA_PIN	GP_BA+0x010	R	GPIO Port A Pin Value Register	0x0000_XXXX
GPIOB_PIN	GP_BA+0x050	R	GPIO Port B Pin Value Register	0x0000_XXXX
GPIOC_PIN	GP_BA+0x090	R	GPIO Port C Pin Value Register	0x0000_XXXX
GPIOD_PIN	GP_BA+0x0D0	R	GPIO Port D Pin Value Register	0x0000_XXXX
GPIOE_PIN	GP_BA+0x110	R	GPIO Port E Pin Value Register	0x0000_XXXX
GPIOF_PIN	GP_BA+0x150	R	GPIO Port F Pin Value Register	0x0000_00XX

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th></tr>
<tr><td colspan="8">PIN</td></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">PIN</td></tr>
</table>

Bits	Description	
[31:16]	**Reserved**	Reserved.
[n] n = 0,1..15	**PIN**	**GPIO Port [X] Pin [N] Value** The value read from each of these bit reflects the actual status of the respective GPI/O pin **Note:** For GPIOE_PIN, bits [15:10] are reserved. For GPIOF_PIN, bits [15:6] are reserved.

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

GPIO Port [A/B/C/D/E/F] De-bounce Enable Register (GPIOx_DBEN)

Register	Offset	R/W	Description	Reset Value
GPIOA_DBEN	GP_BA+0x014	R/W	GPIO Port A De-bounce Enable Register	0x0000_0000
GPIOB_DBEN	GP_BA+0x054	R/W	GPIO Port B De-bounce Enable Register	0x0000_0000
GPIOC_DBEN	GP_BA+0x094	R/W	GPIO Port C De-bounce Enable Register	0x0000_0000
GPIOD_DBEN	GP_BA+0x0D4	R/W	GPIO Port D De-bounce Enable Register	0x0000_0000
GPIOE_DBEN	GP_BA+0x114	R/W	GPIO Port E De-bounce Enable Register	0x0000_0000
GPIOF_DBEN	GP_BA+0x154	R/W	GPIO Port F De-bounce Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DBEN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DBEN							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[n] n = 0,1..15	**DBEN**	**GPIO Port [X] Pin [N] Input Signal De-bounce Enable Control** DBEN[n] used to enable the de-bounce function for each corresponding bit. If the input signal pulse width cannot be sampled by continuous two de-bounce sample cycle the input signal transition is seen as the signal bounce and will not trigger the interrupt. DBEN[n] is used for “edge-trigger” interrupt only, and ignored for “level trigger” interrupt 0 = The GPIO port [x] Pin [n] input signal de-bounce function Disabled. 1 = The GPIO port [x] Pin [n] input signal de-bounce function Enabled. The de-bounce function is valid for edge triggered interrupt. If the interrupt mode is level triggered, the de-bounce enable bit is ignored. **Note:** For GPIOE_DBEN, bits [15:10] are reserved. For GPIOF_DBEN, bits [15:6] are reserved.

GPIO Port [A/B/C/D/E/F] Interrupt Mode Control Register (GPIOx_IMD)

Register	Offset	R/W	Description	Reset Value
GPIOA_IMD	GP_BA+0x018	R/W	GPIO Port A Interrupt Mode Control Register	0x0000_0000
GPIOB_IMD	GP_BA+0x058	R/W	GPIO Port B Interrupt Mode Control Register	0x0000_0000
GPIOC_IMD	GP_BA+0x098	R/W	GPIO Port C Interrupt Mode Control Register	0x0000_0000
GPIOD_IMD	GP_BA+0x0D8	R/W	GPIO Port D Interrupt Mode Control Register	0x0000_0000
GPIOE_IMD	GP_BA+0x118	R/W	GPIO Port E Interrupt Mode Control Register	0x0000_0000
GPIOF_IMD	GP_BA+0x158	R/W	GPIO Port F Interrupt Mode Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
IMD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IMD							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[n] n = 0,1..15	**IMD**	**GPIO Port [X] Pin [N] Edge or Level Detection Interrupt Control** IMD[n] used to control the interrupt is by level trigger or by edge trigger. If the interrupt is by edge trigger, the trigger source is control de-bounce. If the interrupt is by level trigger, the input source is sampled by one clock and the generate the interrupt 0 = Edge trigger interrupt. 1 = Level trigger interrupt. If set pin as the level trigger interrupt, then only one level can be set on the registers GPIOX_IER. If set both the level to trigger interrupt, the setting is ignored and no interrupt will occur The de-bounce function is valid for edge triggered interrupt. If the interrupt mode is level triggered, the de-bounce enable bit is ignored. **Note:** For GPIOE_IMD, bits [15:10] are reserved. For GPIOF_IMD, bits [15:6] are reserved.

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

GPIO Port [A/B/C/D/E/F] Interrupt Enable Register (GPIOx_IER)

Register	Offset	R/W	Description	Reset Value
GPIOA_IER	GP_BA+0x01C	R/W	GPIO Port A Interrupt Enable Register	0x0000_0000
GPIOB_IER	GP_BA+0x05C	R/W	GPIO Port B Interrupt Enable Register	0x0000_0000
GPIOC_IER	GP_BA+0x09C	R/W	GPIO Port C Interrupt Enable Register	0x0000_0000
GPIOD_IER	GP_BA+0x0DC	R/W	GPIO Port D Interrupt Enable Register	0x0000_0000
GPIOE_IER	GP_BA+0x11C	R/W	GPIO Port E Interrupt Enable Register	0x0000_0000
GPIOF_IER	GP_BA+0x15C	R/W	GPIO Port F Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
RIER							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RIER							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FIER							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FIER							

Bits	Description	
[n+16] n = 0,1..15	**RIER**	**GPIO Port [X] Pin [N] Interrupt Enable by Input Rising Edge or Input Level High** RIER[x] used to enable the interrupt for each of the corresponding input GPIO_PIN[x]. Set bit “1” also enable the pin wake-up function When set the RIER[x] bit “1”: If the interrupt is level mode trigger, the input PIN[x] state at level “high” will generate the interrupt. If the interrupt is edge mode trigger, the input PIN[x] state change from “low-to-high” will generate the interrupt. 0 = PIN[x] level-high or low-to-high interrupt Disabled. 1 = PIN[x] level-high or low-to-high interrupt Enabled. **Note:** For GPIOE_IE, bits [31:26] are reserved. For GPIOF_IE, bits [31:22] are reserved.

Bits	Description	
[n] n = 0,1..15	**FIER**	**GPIO Port [X] Pin [N] Interrupt Enable by Input Falling Edge or Input Level Low** FIER[n] used to enable the interrupt for each of the corresponding input GPIO_PIN[n]. Set bit “1” also enable the pin wake-up function When set the FIER[n] bit “1”: If the interrupt is level mode trigger, the input PIN[n] state at level “low” will generate the interrupt. If the interrupt is edge mode trigger, the input PIN[n] state change from “high-to-low” will generate the interrupt. 0 = PIN[n] state low-level or high-to-low change interrupt Disabled. 1 = PIN[n] state low-level or high-to-low change interrupt Enabled. **Note:** For GPIOE_IER, bits [15:10] are reserved. For GPIOF_IER, bits [15:6] are reserved.

GPIO Port [A/B/C/D/E/F] Interrupt Trigger Source Status Register (GPIOx_ISRC)

Register	Offset	R/W	Description	Reset Value
GPIOA_ISRC	GP_BA+0x020	R/W	GPIO Port A Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOB_ISRC	GP_BA+0x060	R/W	GPIO Port B Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOC_ISRC	GP_BA+0x0A0	R/W	GPIO Port C Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOD_ISRC	GP_BA+0x0E0	R/W	GPIO Port D Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOE_ISRC	GP_BA+0x120	R/W	GPIO Port E Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOF_ISRC	GP_BA+0x160	R/W	GPIO Port F Interrupt Trigger Source Status Register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ISRC							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ISRC							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[n] n = 0,1..15	**ISRC**	**GPIO Port [X] Pin [N] Interrupt Trigger Source Indicator** Read Operation: 0 = No interrupt at Port x[n]. 1 = Port x[n] generate an interrupt. Write Operation: 0 = No action. 1 = Clear the correspond pending interrupt. **Note:** For GPIOE_ISRC, bits [15:10] are reserved. For GPIOF_ISRC, bits [15:6] are reserved.

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

GPIO Port [A/B/C/D/E/F] Pull-Up Enable Register (GPIOx_PUEN)

Register	Offset	R/W	Description	Reset Value
GPIOA_PUEN	GP_BA+0x024	R/W	GPIO Port A Pull-Up Enable Register	0x0000_0000
GPIOB_PUEN	GP_BA+0x064	R/W	GPIO Port B Pull-Up Enable Register	0x0000_0000
GPIOC_PUEN	GP_BA+0x0A4	R/W	GPIO Port C Pull-Up Enable Register	0x0000_0000
GPIOD_PUEN	GP_BA+0x0E4	R/W	GPIO Port D Pull-Up Enable Register	0x0000_0000
GPIOE_PUEN	GP_BA+0x124	R/W	GPIO Port E Pull-Up Enable Register	0x0000_0000
GPIOF_PUEN	GP_BA+0x164	R/W	GPIO Port F Pull-Up Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PUEN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PUEN							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[n] n = 0,1..15	**PUEN**	**GPIO Port [X] Pin [N] Pull-up Enable Register** Read Operation: 0 = GPIO port [A/B/C/D/E/F] bit [n] pull-up resistor Disabled. 1 = GPIO port [A/B/C/D/E/F] bit [n] pull-up resistor Enabled. **Note:** For GPIOE_PUEN, bits [15:10] are reserved. For GPIOF_PUEN, bits [15:6] are reserved.

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

De-bounce Cycle Control Register (DBNCECON)

Register	Offset	R/W	Description	Reset Value
DBNCECON	GP_BA+0x180	R/W	De-bounce Cycle Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**DBCLK_ON**	**DBCLKSRC**	**DBCLKSEL**			

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**DBCLK_ON**	**De-bounce Clock Enable Control** This bit controls if the de-bounce clock is enabled. However, if GPI/O pin's interrupt is enabled, the de-bounce clock will be enabled automatically no matter what the DBCLK_ON value is. If CPU is in sleep mode, this bit didn't take effect. And only the GPI/O pin with interrupt enable could get de-bounce clock. 0 = De-bounce clock Disabled. 1 = De-bounce clock Enabled.
[4]	**DBCLKSRC**	**De-bounce Counter Clock Source Selection** 0 = De-bounce counter Clock Source is the HCLK. 1 = De-bounce counter Clock Source is the internal 10 kHz clock.

Bits	Description	
[3:0]	**DBCLKSEL**	**De-bounce Sampling Cycle Selection** 0000 = Sample interrupt input once per 1 clock. 0001 = Sample interrupt input once per 2 clocks. 0010 = Sample interrupt input once per 4 clocks. 0011 = Sample interrupt input once per 8 clocks. 0100 = Sample interrupt input once per 16 clocks. 0101 = Sample interrupt input once per 32 clocks. 0110 = Sample interrupt input once per 64 clocks. 0111 = Sample interrupt input once per 128 clocks. 1000 = Sample interrupt input once per 256 clocks. 1001 = Sample interrupt input once per 2*256 clocks. 1010 = Sample interrupt input once per 4*256clocks. 1011 = Sample interrupt input once per 8*256 clocks. 1100 = Sample interrupt input once per 16*256 clocks. 1101 = Sample interrupt input once per 32*256 clocks. 1110 = Sample interrupt input once per 64*256 clocks. 1111 = Sample interrupt input once per 128*256 clocks.

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

GPIO Port [A/B/C/D/E/F] Bit [n] Data Register (GPIO[A/B/C/D/E/F][n])

Register	Offset	R/W	Description	Reset Value
GPIOA0	GP_BA+0x200	R/W	GPIO Port A Bit 0 Data Register	0x0000_000X
GPIOA1	GP_BA+0x204	R/W	GPIO Port A Bit 1 Data Register	0x0000_000X
GPIOA2	GP_BA+0x208	R/W	GPIO Port A Bit 2 Data Register	0x0000_000X
GPIOA3	GP_BA+0x20C	R/W	GPIO Port A Bit 3 Data Register	0x0000_000X
GPIOA4	GP_BA+0x210	R/W	GPIO Port A Bit 4 Data Register	0x0000_000X
GPIOA5	GP_BA+0x214	R/W	GPIO Port A Bit 5 Data Register	0x0000_000X
GPIOA6	GP_BA+0x218	R/W	GPIO Port A Bit 6 Data Register	0x0000_000X
GPIOA7	GP_BA+0x21C	R/W	GPIO Port A Bit 7 Data Register	0x0000_000X
GPIOA8	GP_BA+0x220	R/W	GPIO Port A Bit 8 Data Register	0x0000_000X
GPIOA9	GP_BA+0x224	R/W	GPIO Port A Bit 9 Data Register	0x0000_000X
GPIOA10	GP_BA+0x228	R/W	GPIO Port A Bit 10 Data Register	0x0000_000X
GPIOA11	GP_BA+0x22C	R/W	GPIO Port A Bit 11 Data Register	0x0000_000X
GPIOA12	GP_BA+0x230	R/W	GPIO Port A Bit 12 Data Register	0x0000_000X
GPIOA13	GP_BA+0x234	R/W	GPIO Port A Bit 13 Data Register	0x0000_000X
GPIOA14	GP_BA+0x238	R/W	GPIO Port A Bit 14 Data Register	0x0000_000X
GPIOA15	GP_BA+0x23C	R/W	GPIO Port A Bit 15 Data Register	0x0000_000X
GPIOB0	GP_BA+0x240	R/W	GPIO Port B Bit 0 Data Register	0x0000_000X
GPIOB1	GP_BA+0x244	R/W	GPIO Port B Bit 1 Data Register	0x0000_000X
GPIOB2	GP_BA+0x248	R/W	GPIO Port B Bit 2 Data Register	0x0000_000X
GPIOB3	GP_BA+0x24C	R/W	GPIO Port B Bit 3 Data Register	0x0000_000X
GPIOB4	GP_BA+0x250	R/W	GPIO Port B Bit 4 Data Register	0x0000_000X
GPIOB5	GP_BA+0x254	R/W	GPIO Port B Bit 5 Data Register	0x0000_000X
GPIOB6	GP_BA+0x258	R/W	GPIO Port B Bit 6 Data Register	0x0000_000X
GPIOB7	GP_BA+0x25C	R/W	GPIO Port B Bit 7 Data Register	0x0000_000X
GPIOB8	GP_BA+0x260	R/W	GPIO Port B Bit 8 Data Register	0x0000_000X
GPIOB9	GP_BA+0x264	R/W	GPIO Port B Bit 9 Data Register	0x0000_000X
GPIOB10	GP_BA+0x268	R/W	GPIO Port B Bit 10 Data Register	0x0000_000X
GPIOB11	GP_BA+0x26C	R/W	GPIO Port B Bit 11 Data Register	0x0000_000X
GPIOB12	GP_BA+0x270	R/W	GPIO Port B Bit 12 Data Register	0x0000_000X

GPIOB13	GP_BA+0x274	R/W	GPIO Port B Bit 13 Data Register	0x0000_000X
GPIOB14	GP_BA+0x278	R/W	GPIO Port B Bit 14 Data Register	0x0000_000X
GPIOB15	GP_BA+0x27C	R/W	GPIO Port B Bit 15 Data Register	0x0000_000X
GPIOC0	GP_BA+0x280	R/W	GPIO Port C Bit 0 Data Register	0x0000_000X
GPIOC1	GP_BA+0x284	R/W	GPIO Port C Bit 1 Data Register	0x0000_000X
GPIOC2	GP_BA+0x288	R/W	GPIO Port C Bit 2 Data Register	0x0000_000X
GPIOC3	GP_BA+0x28C	R/W	GPIO Port C Bit 3 Data Register	0x0000_000X
GPIOC4	GP_BA+0x290	R/W	GPIO Port C Bit 4 Data Register	0x0000_000X
GPIOC5	GP_BA+0x294	R/W	GPIO Port C Bit 5 Data Register	0x0000_000X
GPIOC6	GP_BA+0x298	R/W	GPIO Port C Bit 6 Data Register	0x0000_000X
GPIOC7	GP_BA+0x29C	R/W	GPIO Port C Bit 7 Data Register	0x0000_000X
GPIOC8	GP_BA+0x2A0	R/W	GPIO Port C Bit 8 Data Register	0x0000_000X
GPIOC9	GP_BA+0x2A4	R/W	GPIO Port C Bit 9 Data Register	0x0000_000X
GPIOC10	GP_BA+0x2A8	R/W	GPIO Port C Bit 10 Data Register	0x0000_000X
GPIOC11	GP_BA+0x2AC	R/W	GPIO Port C Bit 11 Data Register	0x0000_000X
GPIOC12	GP_BA+0x2B0	R/W	GPIO Port C Bit 12 Data Register	0x0000_000X
GPIOC13	GP_BA+0x2B4	R/W	GPIO Port C Bit 13 Data Register	0x0000_000X
GPIOC14	GP_BA+0x2B8	R/W	GPIO Port C Bit 14 Data Register	0x0000_000X
GPIOC15	GP_BA+0x2BC	R/W	GPIO Port C Bit 15 Data Register	0x0000_000X
GPIOD0	GP_BA+0x2C0	R/W	GPIO Port D Bit 0 Data Register	0x0000_000X
GPIOD1	GP_BA+0x2C4	R/W	GPIO Port D Bit 1 Data Register	0x0000_000X
GPIOD2	GP_BA+0x2C8	R/W	GPIO Port D Bit 2 Data Register	0x0000_000X
GPIOD3	GP_BA+0x2CC	R/W	GPIO Port D Bit 3 Data Register	0x0000_000X
GPIOD4	GP_BA+0x2D0	R/W	GPIO Port D Bit 4 Data Register	0x0000_000X
GPIOD5	GP_BA+0x2D4	R/W	GPIO Port D Bit 5 Data Register	0x0000_000X
GPIOD6	GP_BA+0x2D8	R/W	GPIO Port D Bit 6 Data Register	0x0000_000X
GPIOD7	GP_BA+0x2DC	R/W	GPIO Port D Bit 7 Data Register	0x0000_000X
GPIOD8	GP_BA+0x2E0	R/W	GPIO Port D Bit 8 Data Register	0x0000_000X
GPIOD9	GP_BA+0x2E4	R/W	GPIO Port D Bit 9 Data Register	0x0000_000X
GPIOD10	GP_BA+0x2E8	R/W	GPIO Port D Bit 10 Data Register	0x0000_000X
GPIOD11	GP_BA+0x2EC	R/W	GPIO Port D Bit 11 Data Register	0x0000_000X

GPIOD12	GP_BA+0x2F0	R/W	GPIO Port D Bit 12 Data Register	0x0000_000X
GPIOD13	GP_BA+0x2F4	R/W	GPIO Port D Bit 13 Data Register	0x0000_000X
GPIOD14	GP_BA+0x2F8	R/W	GPIO Port D Bit 14 Data Register	0x0000_000X
GPIOD15	GP_BA+0x2FC	R/W	GPIO Port D Bit 15 Data Register	0x0000_000X
GPIOE0	GP_BA+0x300	R/W	GPIO Port E Bit 0 Data Register	0x0000_000X
GPIOE1	GP_BA+0x304	R/W	GPIO Port E Bit 1 Data Register	0x0000_000X
GPIOE2	GP_BA+0x308	R/W	GPIO Port E Bit 2 Data Register	0x0000_000X
GPIOE3	GP_BA+0x30C	R/W	GPIO Port E Bit 3 Data Register	0x0000_000X
GPIOE4	GP_BA+0x310	R/W	GPIO Port E Bit 4 Data Register	0x0000_000X
GPIOE5	GP_BA+0x314	R/W	GPIO Port E Bit 5 Data Register	0x0000_000X
GPIOE6	GP_BA+0x318	R/W	GPIO Port E Bit 6 Data Register	0x0000_000X
GPIOE7	GP_BA+0x31C	R/W	GPIO Port E Bit 7 Data Register	0x0000_000X
GPIOE8	GP_BA+0x320	R/W	GPIO Port E Bit 8 Data Register	0x0000_000X
GPIOE9	GP_BA+0x324	R/W	GPIO Port E Bit 9 Data Register	0x0000_000X
GPIOF0	GP_BA+0x340	R/W	GPIO Port F Bit 0 Data Register	0x0000_000X
GPIOF1	GP_BA+0x344	R/W	GPIO Port F Bit 1 Data Register	0x0000_000X
GPIOF2	GP_BA+0x348	R/W	GPIO Port F Bit 2 Data Register	0x0000_000X
GPIOF3	GP_BA+0x34C	R/W	GPIO Port F Bit 3 Data Register	0x0000_000X
GPIOF4	GP_BA+0x350	R/W	GPIO Port F Bit 4 Data Register	0x0000_000X
GPIOF5	GP_BA+0x354	R/W	GPIO Port F Bit 5 Data Register	0x0000_000X

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**GPIO[x][n]**

Bits	Description	
[31:1]	**Reserved**	Reserved.

Bits	Description	
[0]	**GPIO[x][n]**	**GPIO Port [X] Pin [N] I/O Data** This field supports the bit operation mode on related GPIO port [x] pin [n]. Writing this filed to set the corresponding GPIO port [x] pin [n] output value while reading this field to get the corresponding GPIO port [x] pin [n] value. Read Operation: 0 = The corresponding GPIO port [x] pin [n] value is low. 1 = The corresponding GPIO port [x] pin [n] value is high. Write Operation: 0 = Set corresponding GPIO port [x] pin [n] to low. 1 = Set corresponding GPIO port [x] pin [n] to high. **Note:** The write operation will not be affected by register GPIOx_DMASK.

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

6.8 DMA Controller

6.8.1 Overview

The NuMicro® NANO112 series DMA contains four-channel peripheral direct memory access (PDMA) controller and a cyclic redundancy check (CRC) generator.

The PDMA that transfers data to and from memory or transfer data to and from peripherals. For PDMA channel (PDMA CH1~CH4), there is one-word buffer as transfer buffer between the Peripherals APB devices and Memory. User can stop the PDMA operation by disable PDMACEN (PDMA_CSRx[0]). User can polling TD_IS (PDMA_ISRx[1]) or enable BLKD_IE (PDMA_IERx[1]) and wait interrupt to check PDMA transfer complete . The DMA controller can increase source or destination address, fixed or wrap around them as well.

The DMA controller contains a cyclic redundancy check (CRC) generator that can perform CRC calculation with programmable polynomial settings. The CRC engine supports CPU mode and DMA transfer mode.

6.8.2 Features

- Supports four PDMA channels (CH1 ~ CH4) and one CRC channel. Each PDMA channel can support a unidirectional transfer
- AMBA AHB master/slave interface compatible, for data transfer and register read/write
- Hardware round robin priority scheme. DMA channel 1 has the highest priority and channel 4 has the lowest priority
- PDMA
  - Peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfer
  - Supports word boundary address
  - Supports word alignment transfer length in memory-to-memory mode
  - Supports word/half-word/byte alignment transfer length in peripheral-to-memory and memory-to-peripheral mode
  - Supports word/half-word/byte transfer data width from/to peripheral
  - Supports address direction: increment, fixed, and wrap around
  - Supports time-out function in all channel
- Cyclic Redundancy Check (CRC)
  - Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and CRC-32
    - CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
    - CRC-8: $X^8 + X^2 + X + 1$
    - CRC-16: $X^{16} + X^{15} + X^2 + 1$
    - CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
  - Programmable seed value
  - Supports programmable order reverse setting for input data and CRC checksum
  - Supports programmable 1’s complement setting for input data and CRC checksum
  - Supports CPU mode or DMA transfer mode

- Supports 8/16/32-bit of data width in CRC CPU mode
  - 8-bit write mode: 1-AHB clock cycle operation
  - 16-bit write mode: 2-AHB clock cycle operation
  - 32-bit write mode: 4-AHB clock cycle operation
- Supports byte alignment transfer length in CRC DMA mode

6.8.3 Block Diagram

The DMA clock control and block diagram are shown as follows.

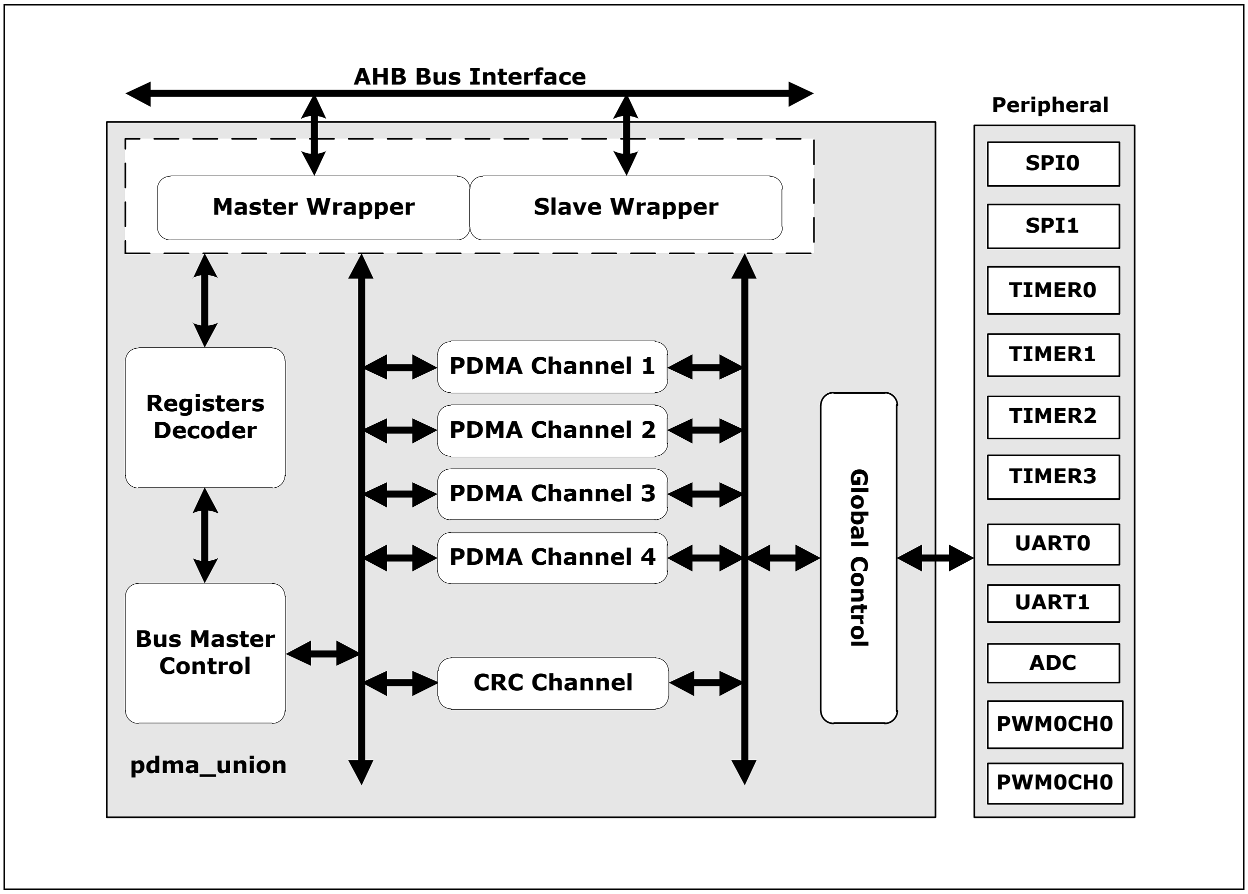


Figure 6-25 DMA Controller Block Diagram

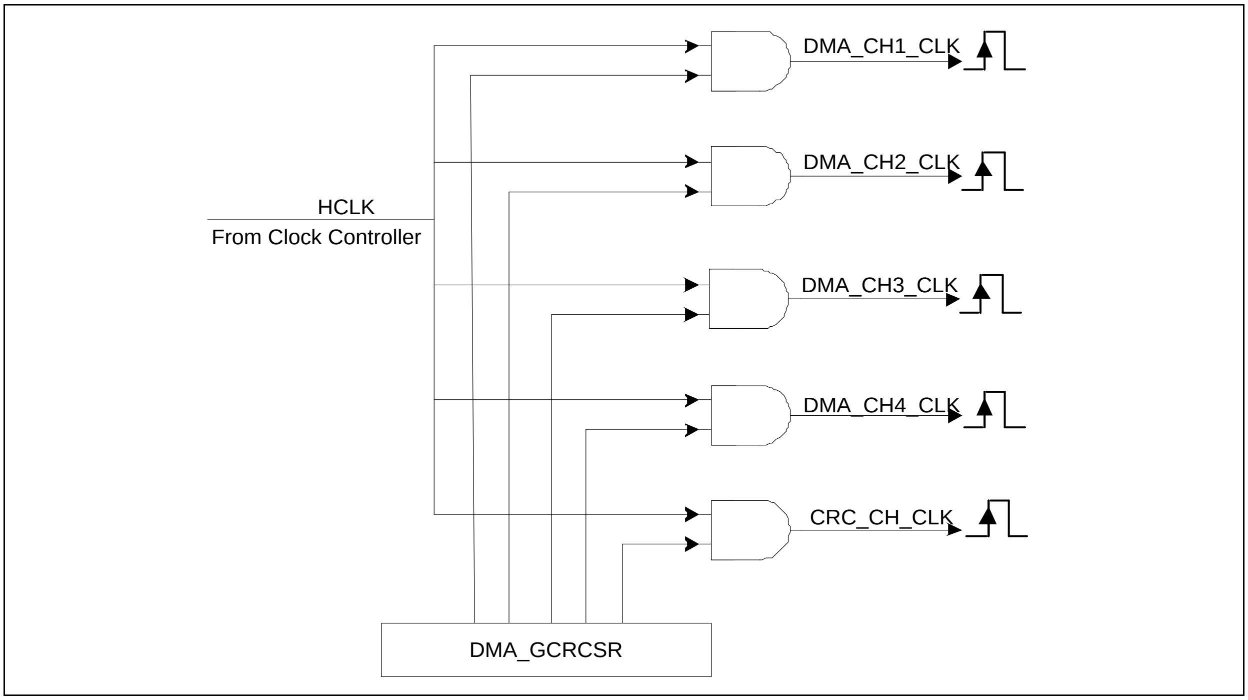


Figure 6-26 DMA Clock Control Diagram

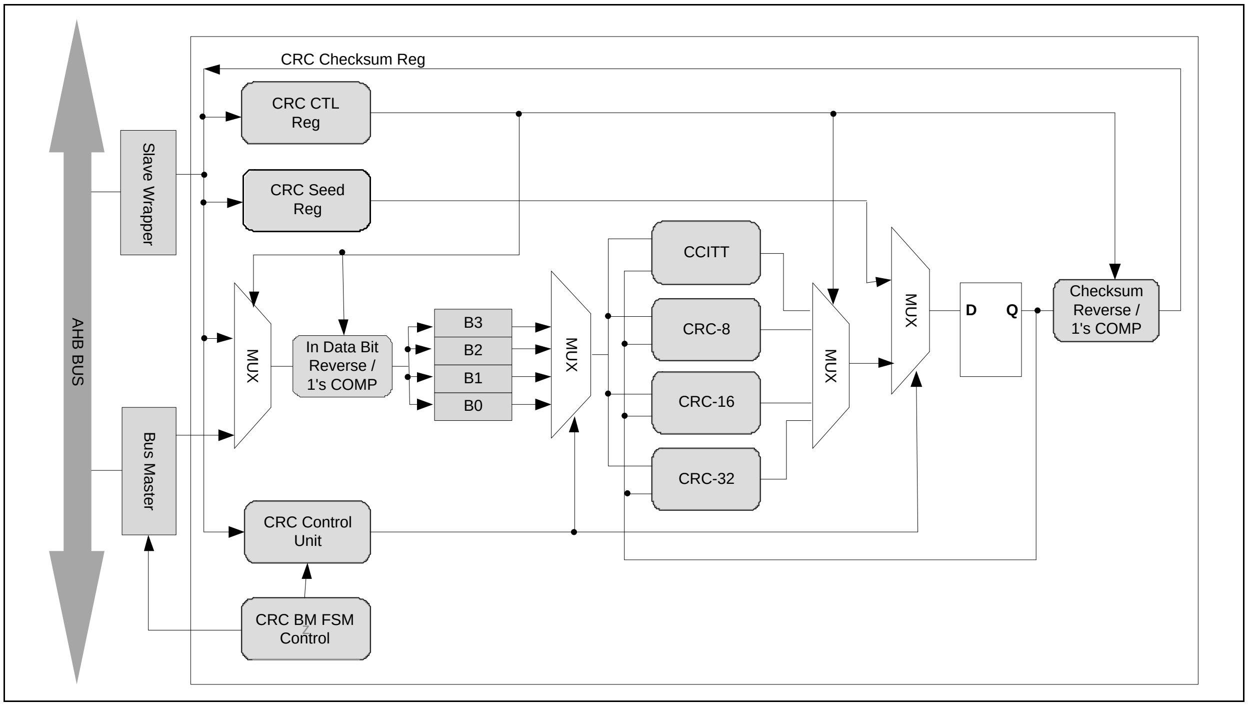


Figure 6-27 CRC Generator Block Diagram

6.8.4 Functional Description

The direct memory access (DMA) controller module transfers data from one address to another address, without CPU intervention. The DMA controller contains four PDMA (Peripheral-to-Memory or Memory-to-Peripheral or Memory-to-Memory) channels and one CRC generator channel.

User can recognize the completion of a DMA operation by polling or when it receives an internal DMA interrupt. As to the source and destination address, the DMA controller has three different modes: increased, fixed and wrap around operation mode.

6.8.4.1 PDMA

The DMA controller has four channels PDMA (Peripheral-to-Memory or Memory-to-Peripheral or Memory-to-Memory). As to the source and destination address, the DMA controller has three different modes: increased, fixed and wrap around operation mode.

Every PDMA channel behavior is not pre-defined, users must configure the channel service settings of DMA_DSSR0, DMA_DSSR1 registers before starting the related PDMA channel.

User must enable DMA channel PDMACEN (PDMA_CSRx[0]) and then write a valid source address to the PDMA_SARx register, a destination address to the PDMA_DARx register, and a transfer count to the PDMA_BCRx register. Next, trigger the TRIG_EN (PDMA_CSRx[23]). If the source address and destination is not in wrap around mode, the PDMA will continue the transfer until PDMA_CBCRx comes down to 0 (in wrap around mode, when PDMA_CBCRx is equal to 0, the PDMA will reload PDMA_CBCRx and work around until user disables PDMACEN (PDMA_CSRx[0])).

The following sequence is a program sequence example.

- Enable PDMA Channel engine clock by setting CLKx_EN (DMA_GCRCSR [9~12]) register.
- Configure the channel service setting by setting DMA_DSSR0 / DMA_DSSR1 register.
- Configure PDMA_CSRx register:
    - Enable PDMA channel (PDMACEN)
    - Set source/destination address direction (SAD_SEL / DAD_SEL)
    - Configure PDMA mode selection (MODE_SEL (PDMA_CSRx[3:2]))
    - Configure peripheral transfer width selection (APB_TWS(PDMA_CSRx[20:19])).
- Configure source /destination address by setting PDMA_SARx/PDMA_DARx registers.
- Configure PDMA_transfer byte count by setting PDMA_BCRx register.
- Enable PDMA block transfer done interrupt by setting TD_IE (PDMA_IERx[1]). (optional)
- Enable PDMA NVIC by setting NVIC_ISER register bit 26 to “1”. (optional)
- Enable PDMA read/write transfer by setting TRIG_EN (PDMA_CSRx[23]) register.
- If PDMA block transfer done interrupt is generated, write “1” to TD_IS (PDMA_ISRx[1]) by user to clear interrupt flag.
- Enable PDMA read/write transfer by setting the TRIG_EN for the next block transfer.

If an error occurs during the PDMA operation, the channel stops unless user clears the error condition and sets the SW_RST (PDMA_CSRx [1]) to reset the PDMA channel and set PDMACEN and TRIG_EN to start again.

In PDMA (Peripheral-to-Memory or Memory-to-Peripheral) mode, DMA can transfer data between the

Peripherals APB IP (ex: UART, SPI, ADC….) and Memory.

6.8.4.2 CRC

The DMA controller contains a cyclic redundancy check (CRC) generator that can perform CRC calculation with programmable polynomial settings. The operation polynomial includes CRC-CCITT, CRC-8, CRC-16 and CRC-32; User can choose the CRC operation polynomial mode by setting CRC_MODE[1:0] (CRC_CTL[31:30] CRC Polynomial Mode).

The CRC engine supports CPU mode if CRCCEN bit (CRC_CLT [0] CRC Channel Enable) is 1, TRIG_EN bit (CRC_CTL [23] CRC DMA Trigger Enable) is 0 and DMA transfer mode if CRCCEN bit (CRC_CLT [0] CRC Channel Enable) is 1, TRIG_EN bit (CRC_CTL [23] CRC DMA Trigger Enable) is 1. The following sequence is a program sequence example.

The procedure when operating in CPU mode:

- Enable CRC engine by setting CRCCEN bit (CRC_CLT [0] CRC Channel Enable) to 1.
- Set the transfer data format WDATA_RVS (CRC_CTL [24] Write Data Order Reverse), CHECKSUM_RVS (CRC_CTL [25] Checksum Reverse), WDATA_COM (CRC_CTL [26] Write Data 1's Complement), CHECKSUM_COM (CRC_CTL [27] Checksum 1's Complement), initial seed value in CRC_SEED (CRC_SEED [31:0] CRC Seed Register) and select write data length by setting CPU_WDLEN [1:0] (CRC_CTL [29:28] CPU Write Data Length).
- Set the CRC_RST bit (CRC_CTL [1] CRC Engine Reset) to 1 to load the initial seed value to CRC circuit but others contents of CRT_CTL register will not be cleared. This bit will be cleared automatically.
- Write data to CRC_WDATA (CRC_WDATA [31:0] CRC Write Data Register) to perform CRC calculation.
- Then, get the CRC checksum results by reading the CRC_CHECKSUM (CRC_CHECKSUM [31:0] CRC Checksum Register).

Procedure when operating in CRC DMA mode:

- Enable CRC engine by setting CRCCEN bit (CRC_CLT [0] CRC Channel Enable) to 1.
- Set the transfer data format WDATA_RVS (CRC_CTL [24] Write Data Order Reverse), CHECKSUM_RVS (CRC_CTL [25] Checksum Reverse), WDATA_COM (CRC_CTL [26] Write Data 1's Complement), CHECKSUM_COM (CRC_CTL [27] Checksum 1's Complement) and initial seed value in CRC_SEED (CRC_SEED [31:0] CRC Seed Register).
- Specify a valid source address (word alignment) and transfer counts by setting CRC_DMASAR (CRC_DMASAR[31:0] CRC DMA Transfer Source Address Register) and CRC_DMABCR (CRC_DMABCR [15:0] CRC DMA Transfer Byte Count Register).
- Set TRIG_EN bit (CRC_CTL [23] CRC DMA Trigger Enable) to 1 to perform CRC calculation.
- Wait CRC DMA transfer and check if CRC DMA transfer is done by the CRC_BLKD_IF bit (CRC DMA Block Transfer Done Interrupt Flag), and then get the CRC checksum results by reading the CRC_CHECKSUM (CRC_CHECKSUM [31:0] CRC Checksum Register).

6.8.5 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
PDMA Base Address: **PDMA_CHx_BA = 0x5000_8000 + (0x100 * x)** **x = 1,2..4** **CRC_BA = 0x5000_8E00** **DMA_GCR_BA = 0x5000_8F00**				
PDMA_CSR	PDMA_CHx_BA+0x00	R/W	PDMA Control Register	0x0000_0000
PDMA_SAR	PDMA_CHx_BA+0x04	R/W	PDMA Source Address Register	0x0000_0000
PDMA_DAR	PDMA_CHx_BA+0x08	R/W	PDMA Destination Address Register	0x0000_0000
PDMA_BCR	PDMA_CHx_BA+0x0C	R/W	PDMA Transfer Byte Count Register	0x0000_0000
PDMA_CSAR	PDMA_CHx_BA+0x14	R	PDMA Current Source Address Register	0x0000_0000
PDMA_CDAR	PDMA_CHx_BA+0x18	R	PDMA Current Destination Address Register	0x0000_0000
PDMA_CBCR	PDMA_CHx_BA+0x1C	R	PDMA Current Transfer Byte Count Register	0x0000_0000
PDMA_IER	PDMA_CHx_BA+0x20	R/W	PDMA Interrupt Enable Register	0x0000_0001
PDMA_ISR	PDMA_CHx_BA+0x24	R/W	PDMA Interrupt Status Register	0x0000_0000
PDMA_TCR	PDMA_CHx_BA+0x28	R/W	PDMA Timer Counter Setting Register	0x0000_0000
CRC_CTL	CRC_BA+0x00	R/W	CRC Control Register	0x2000_0000
CRC_DMASAR	CRC_BA+0x04	R/W	CRC DMA Source Address Register	0x0000_0000
CRC_DMABCR	CRC_BA+0x0C	R/W	CRC DMA Transfer Byte Count Register	0x0000_0000
CRC_DMACSAR	CRC_BA+0x14	R	CRC DMA Current Source Address Register	0x0000_0000
CRC_DMACBCR	CRC_BA+0x1C	R	CRC DMA Current Transfer Byte Count Register	0x0000_0000
CRC_DMAIER	CRC_BA+0x20	R/W	CRC DMA Interrupt Enable Register	0x0000_0001
CRC_DMAISR	CRC_BA+0x24	R/W	CRC DMA Interrupt Status Register	0x0000_0000
CRC_WDATA	CRC_BA+0x80	R/W	CRC Write Data Register	0x0000_0000
CRC_SEED	CRC_BA+0x84	R/W	CRC Seed Register	0xFFFF_FFFF
CRC_CHECKSUM	CRC_BA+0x88	R	CRC Checksum Register	0xFFFF_FFFF
DMA_GCRCSR	DMA_GCR_BA+0x00	R/W	DMA Global Control Register	0x0000_0000
DMA_DSSR0	DMA_GCR_BA+0x04	R/W	DMA Service Selection Control Register 0	0x1F1F_1F00
DMA_DSSR1	DMA_GCR_BA+0x08	R/W	DMA Service Selection Control Register 1	0x0000_001F
DMA_GCRISR	DMA_GCR_BA+0x0C	R	DMA Global Interrupt Status Register	0x0000_0000

Note: The x of the PDMA_REGx represents the PDMA channel.

6.8.6 Register Description

PDMA Control Register (PDMA_CSR)

Register	Offset	R/W	Description	Reset Value
PDMA_CSR	PDMA_CHx_BA+0x00	R/W	PDMA Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TRIG_EN	Reserved		APB_TWS		Reserved		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			TO_EN	Reserved			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DAD_SEL		SAD_SEL		MODE_SEL		SW_RST	PDMACEN

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23]	**TRIG_EN**	**Trigger Enable Control** 0 = No effect. 1 = PDMA data read or write transfer Enabled. **Note:** When PDMA transfer completed, this bit will be cleared automatically. If the bus error occurs, all PDMA transfer will be stopped. User must reset all PDMA channel, and then trigger again.
[22:21]	**Reserved**	Reserved.
[20:19]	**APB_TWS**	**Peripheral Transfer Width Selection** 00 = One word (32-bit) is transferred for every PDMA operation. 01 = One byte (8-bit) is transferred for every PDMA operation. 10 = One half-word (16-bit) is transferred for every PDMA operation. 11 = Reserved. **Note:** This field is meaningful only when MODE_SEL is Peripheral to Memory mode (Peripheral-to-Memory) or Memory to Peripheral mode (Memory-to-Peripheral).
[18:13]	**Reserved**	Reserved.
[12]	**TO_EN**	**Time-out Enable Control** This bit will enable PDMA internal counter. While this counter counts to 0, the TO_IS will be set. 0 = PDMA internal counter Disabled. 1 = PDMA internal counter Enabled.
[11:8]	**Reserved**	Reserved.
[7:6]	**DAD_SEL**	**Transfer Destination Address Direction Selection** 00 = Transfer Destination address is incremented successively. 01 = Reserved.

Bits	Description	
		10 = Transfer Destination address is fixed (This feature can be used when data where transferred from multiple sources to a single destination). 11 = Transfer Destination address is wrapped around (When the PDMA_CBCR is equal to 0, the PDMA_CDAR and PDMA_CBCR register will be updated by PDMA_DAR and PDMA_BCR automatically. PDMA will start another transfer without user trigger until PDMA_EN disabled. When the PDMA_EN is disabled, the PDMA will complete the active transfer but the remained data which in the PDMA_BUF will not transfer to destination address).
[5:4]	**SAD_SEL**	**Transfer Source Address Direction Selection** 00 = Transfer Source address is incremented successively. 01 = Reserved. 10 = Transfer Source address is fixed (This feature can be used when data where transferred from a single source to multiple destinations). 11 = Transfer Source address is wrap around (When the PDMA_CBCR is equal to 0, the PDMA_CSAR and PDMA_CBCR register will be updated by PDMA_SAR and PDMA_BCR automatically. PDMA will start another transfer without user trigger until PDMA_EN disabled. When the PDMA_EN is disabled, the PDMA will complete the active transfer but the remained data which in the PDMA_BUF will not transfer to destination address).
[3:2]	**MODE_SEL**	**PDMA Mode Selection** 00 = Memory to Memory mode (Memory-to-Memory). 01 = Peripheral to Memory mode (Peripheral-to-Memory). 10 = Memory to Peripheral mode (Memory-to-Peripheral). 11 = Reserved.
[1]	**SW_RST**	**Software Engine Reset** 0 = No effect. 1 = Reset the internal state machine, pointers and internal buffer. The contents of control register will not be cleared. This bit will be automatically cleared after few clock cycles.
[0]	**PDMACEN**	**PDMA Channel Enable Control** Setting this bit to 1 enables PDMA operation. If this bit is cleared, PDMA will ignore all PDMA request and force Bus Master into IDLE state. **Note:** SW_RST(PDMA_CSRx[1], x= 1~4) will clear this bit.

PDMA Source Address Register (PDMA_SAR)

Register	Offset	R/W	Description	Reset Value
PDMA_SAR	PDMA_CHx_BA+0x04	R/W	PDMA Source Address Register	0x0000_0000

31	30	29	28	27	26	25	24
PDMA_SAR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PDMA_SAR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PDMA_SAR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDMA_SAR							

Bits	Description	
[31:0]	**PDMA_SAR**	**PDMA Transfer Source Address Bits** This field indicates a 32-bit source address of PDMA. **Note:** The source address must be word alignment.

PDMA Destination Address Register (PDMA_DAR)

Register	Offset	R/W	Description	Reset Value
PDMA_DAR	PDMA_CHx_BA+0x08	R/W	PDMA Destination Address Register	0x0000_0000

31	30	29	28	27	26	25	24
PDMA_DAR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PDMA_DAR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PDMA_DAR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDMA_DAR							

Bits	Description	
[31:0]	**PDMA_DAR**	**PDMA Transfer Destination Address Bits** This field indicates a 32-bit destination address of PDMA. **Note:** The destination address must be word alignment

PDMA Transfer Byte Count Register (PDMA_BCR)

Register	Offset	R/W	Description	Reset Value
PDMA_BCR	PDMA_CHx_BA+0x0C	R/W	PDMA Transfer Byte Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PDMA_BCR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDMA_BCR							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**PDMA_BCR**	**PDMA Transfer Byte Count Bits** This field indicates a 16-bit transfer byte count number of PDMA; it must be word alignment.

PDMA Current Source Address Register (PDMA_CSAR)

Register	Offset	R/W	Description	Reset Value
PDMA_CSAR	PDMA_CHx_BA+0x14	R	PDMA Current Source Address Register	0x0000_0000

31	30	29	28	27	26	25	24
PDMA_CSAR							
23	22	21	20	19	18	17	16
PDMA_CSAR							
15	14	13	12	11	10	9	8
PDMA_CSAR							
7	6	5	4	3	2	1	0
PDMA_CSAR							

Bits	Description	
[31:0]	**PDMA_CSAR**	**PDMA Current Source Address Bits (Read Only)** This field indicates the source address where the PDMA transfer just occurred.

PDMA Current Destination Address Register (PDMA_CDAR)

Register	Offset	R/W	Description	Reset Value
PDMA_CDAR	PDMA_CHx_BA+0x18	R	PDMA Current Destination Address Register	0x0000_0000

31	30	29	28	27	26	25	24
PDMA_CDAR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PDMA_CDAR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PDMA_CDAR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDMA_CDAR							

Bits	Description	
[31:0]	**PDMA_CDAR**	**PDMA Current Destination Address Bits (Read Only)** This field indicates the destination address where the PDMA transfer just occurred.

PDMA Current Byte Count Register (PDMA_CBCR)

Register	Offset	R/W	Description	Reset Value
PDMA_CBCR	PDMA_CHx_BA+0x1C	R	PDMA Current Transfer Byte Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PDMA_CBCR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDMA_CBCR							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**PDMA_CBCR**	**PDMA Current Byte Count Bits (Read Only)** This field indicates the current remained byte count of PDMA. **Note1:** This field value will be cleared to 0 when user sets SW_RST (PDMA_CSRx[1]) to "1".

PDMA Interrupt Enable Control Register (PDMA_IER)

Register	Offset	R/W	Description	Reset Value
PDMA_IER	PDMA_CHx_BA+0x20	R/W	PDMA Interrupt Enable Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**TO_IE**	**WRA_BCR_IE**				**TD_IE**	**TABORT_IE**

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**TO_IE**	**Time-out Interrupt Enable Control** 0 = Time-out interrupt Disabled. 1 = Time-out interrupt Enabled.
[5:2]	**WRA_BCR_IE**	**Wrap Around Byte Count Interrupt Enable Control** 0001 = Interrupt enable of PDMA_CBCR equals 0. 0100 = Interrupt enable of PDMA_CBCR equals 1/2 PDMA_BCR. Others = Reserved.
[1]	**TD_IE**	**PDMA Block Transfer Done Interrupt Enable Control** 0 = Interrupt generator Disabled when PDMA transfer is done. 1 = Interrupt generator Enabled when PDMA transfer is done.
[0]	**TABORT_IE**	**PDMA Read/Write Target Abort Interrupt Enable Control** 0 = Target abort interrupt generation Disabled during PDMA transfer. 1 = Target abort interrupt generation Enabled during PDMA transfer.

PDMA Interrupt Status Register (PDMA_ISR)

Register	Offset	R/W	Description	Reset Value
PDMA_ISR	PDMA_CHx_BA+0x24	R/W	PDMA Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	TO_IS	WRA_BCR_IS				TD_IS	TABORT_IS

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**TO_IS**	**Time-out Interrupt Status Flag** This flag indicated that PDMA has waited peripheral request for a period defined by PDMA_TCR. 0 = No time-out flag. 1 = Time-out flag. **Note:** This bit is cleared by writing "1" to it.
[5:2]	**WRA_BCR_IS**	**Wrap Around Transfer Byte Count Interrupt Status Flag** WAR_BCR_IS [0] (xxx1) = PDMA_CBCR equal 0 flag. WAR_BCR_IS [2] (x1xx) = PDMA_CBCR equal 1/2 PDMA_BCR flag. **Note:** Each bit is cleared by writing "1" to it. This field is only valid in wrap around mode. (DAD_SEL (PDMA_CSR[7:6]) =11 or SAD_SEL (PDMA_CSR[5:4]) =11).
[1]	**TD_IS**	**Transfer Done Interrupt Status Flag** This bit indicates that PDMA has finished all transfer. 0 = Not finished yet. 1 = Done. **Note:** This bit is cleared by writing "1" to it.
[0]	**TABORT_IS**	**PDMA Read/Write Target Abort Interrupt Status Flag** 0 = No bus ERROR response received. 1 = Bus ERROR response received. **Note1:** This bit is cleared by writing "1" to it. **Note2:** This bit indicates bus master received ERROR response or not, if bus master received occur it means that target abort is happened. PDMA controller will stop transfer and respond this event to user then go to IDLE state. When target abort occurred, user must reset PDMA controller, and then transfer those data again.

PDMA Timer Count Setting Register (PDMA_TCR)

Register	Offset	R/W	Description	Reset Value
PDMA_TCR	PDMA_CHx_BA+0x28	R/W	PDMA Timer Counter Setting Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PDMA_TCR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDMA_TCR							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**PDMA_TCR**	**PDMA Timer Count Setting** Each PDMA channel contains an internal counter. This internal counter will reload and start counting when completing each peripheral request service. The internal counter loads the value of PDAM_TCR and starts counting down when setting TO_EN (PDMA_CSRx[12]). PDMA will request interrupt when this internal counter reaches 0 and TO_IE (PDMA_IERx[6]) is 1.

CRC Control Register (CRC_CTL)

Register	Offset	R/W	Description	Reset Value
CRC_CTL	CRC_BA+0x00	R/W	CRC Control Register	0x2000_0000

31	30	29	28	27	26	25	24
CRC_MODE		CPU_WDLEN		CHECKSUM_ COM	WDATA_COM	CHECKSUM_ RVS	WDATA_RVS
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TRIG_EN	Reserved						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						CRC_RST	CRCCEN

Bits	Description	
[31:30]	**CRC_MODE**	**CRC Polynomial Mode** This field indicates the CRC operation polynomial mode. 00 = CRC-CCITT Polynomial Mode. 01 = CRC-8 Polynomial Mode. 10 = CRC-16 Polynomial Mode. 11 = CRC-32 Polynomial Mode.
[29:28]	**CPU_WDLEN**	**CPU Write Data Length** This field indicates the CPU write data length only when operating in CPU mode. 00 = The write data length is 8-bit mode. 01 = The write data length is 16-bit mode. 10 = The write data length is 32-bit mode. 11 = Reserved. **Note1:** This field is only valid when operating in CPU mode. **Note2:** When the write data length is 8-bit mode, the valid data in CRC_WDATA register is only CRC_WDATA [7:0] bits; if the write data length is 16-bit mode, the valid data in CRC_WDATA register is only CRC_WDATA [15:0].
[27]	**CHECKSUM_COM**	**Checksum 1’s Complement** This bit is used to enable the 1’s complement function for checksum result in CRC_CHECKSUM register. 0 = 1’s complement for CRC checksum Disabled. 1 = 1’s complement for CRC checksum Enabled.
[26]	**WDATA_COM**	**Write Data 1’s Complement** This bit is used to enable the 1’s complement function for write data value in CRC_WDTAT register. 0 = 1’s complement for CRC write data in Disabled. 1 = 1’s complement for CRC write data in Enabled.

[25]	**CHECKSUM_RVS**	**Checksum Reverse** This bit is used to enable the bit order reverse function for write data value in CRC_CHECKSUM register. 0 = Bit order reverse for CRC checksum Disabled. 1 = Bit order reverse for CRC checksum Enabled. **Note:** If the checksum result is 0XDD7B0F2E, the bit order reverse for CRC checksum is 0x74F0DEBB
[24]	**WDATA_RVS**	**Write Data Order Reverse** This bit is used to enable the bit order reverse function for write data value in CRC_WDTAT register. 0 = Bit order reverse for CRC write data in Disabled. 1 = Bit order reverse for CRC write data in Enabled (per byre). **Note:** If the write data is 0xAABBCCDD, the bit order reverse for CRC write data in is 0x55DD33BB
[23]	**TRIG_EN**	**Trigger Enable Control** This bit is used to trigger the CRC DMA transfer. 0 = No effect. 1 = CRC DMA data read or write transfer Enabled. **Note1:** If this bit asserts which indicates the CRC engine operation in CRC DMA mode, do not fill in any data in CRC_WDATA register. **Note2:** When CRC DMA transfer completed, this bit will be cleared automatically. **Note3:** If the bus error occurs when CRC DMA transfer data, all CRC DMA transfer will be stopped. User must reset all DMA channel before trigger DMA again.
[22:2]	**Reserved**	Reserved.
[1]	**CRC_RST**	**CRC Engine Reset** 0 = No effect. 1 = Reset the internal CRC state machine and internal buffer. The others contents of CRC_CTL register will not be cleared. This bit will be cleared automatically. **Note:** When operated in CPU mode, setting this bit will reload the initial seed value (CRC_SEED register).
[0]	**CRCCEN**	**CRC Channel Enable Control** 0 = No effect. 1 = CRC operation Enabled. When operating in CRC DMA mode (TRIG_EN = 1), if user clears this bit, the DMA operation will be continuous until all CRC DMA operation is done, and the TRIG_EN bit will keep 1until all CRC DMA operation done. But in this case, the CRC_DMAISR [BLKD_IF] flag will inactive, user can read CRC checksum result only if TRIG_EN clears to 0 When operating in CRC DMA mode (TRIG_EN = 1), if user wants to stop the transfer immediately, user can write 1 to CRC_RST bit (CRC_CTL [1] CRC Engine Reset) to stop the transmission.

CRC DMA Source Address Register (CRC_DMASAR)

Register	Offset	R/W	Description	Reset Value
CRC_DMASAR	CRC_BA+0x04	R/W	CRC DMA Source Address Register	0x0000_0000

31	30	29	28	27	26	25	24
CRC_DMASAR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CRC_DMASAR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CRC_DMASAR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRC_DMASAR							

Bits	Description	
[31:0]	**CRC_DMASAR**	**CRC DMA Transfer Source Address Bits** This field indicates a 32-bit source address of CRC DMA. (CRC_DMASAR + CRC_DMABCR) = (CRC_DMACSAR + CRC_DMACBCR). **Note:** The source address must be word alignment

CRC DMA Transfer Byte Count Register (CRC_DMABCR)

Register	Offset	R/W	Description	Reset Value
CRC_DMABCR	CRC_BA+0x0C	R/W	CRC DMA Transfer Byte Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CRC_DMABCR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRC_DMABCR							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CRC_DMABCR**	**CRC DMA Transfer Byte Count** This field indicates a 16-bit total transfer byte count number of CRC DMA (CRC_DMASAR + CRC_DMABCR) = (CRC_DMACSAR + CRC_DMACBCR).

CRC DMA Current Source Address Register (CRC_DMACSAR)

Register	Offset	R/W	Description	Reset Value
CRC_DMACSAR	CRC_BA+0x14	R	CRC DMA Current Source Address Register	0x0000_0000

31	30	29	28	27	26	25	24
CRC_DMACSAR							
23	22	21	20	19	18	17	16
CRC_DMACSAR							
15	14	13	12	11	10	9	8
CRC_DMACSAR							
7	6	5	4	3	2	1	0
CRC_DMACSAR							

Bits	Description	
[31:0]	**CRC_DMACSAR**	**CRC DMA Current Source Address Bits (Read Only)** This field indicates the current source address where the CRC DMA transfer just occurs. (CRC_DMASAR + CRC_DMABCR) = (CRC_DMACSAR + CRC_DMACBCR).

CRC DMA Current Transfer Byte Count Register (CRC_DMACBCR)

Register	Offset	R/W	Description	Reset Value
CRC_DMACBCR	CRC_BA+0x1C	R	CRC DMA Current Transfer Byte Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CRC_DMACBCR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRC_DMACBCR							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CRC_DMACBCR**	**CRC DMA Current Remained Byte Count (Read Only)** This field indicates the current remained byte count of CRC DMA. (CRC_DMASAR + CRC_DMABCR) = (CRC_DMACSAR + CRC_DMACBCR). **Note:** Setting the CRC_RST bit to 1 will clear this register value.

CRC DMA Interrupt Enable Register (CRC_DMAIER)

Register	Offset	R/W	Description	Reset Value
CRC_DMAIER	CRC_BA+0x20	R/W	CRC DMA Interrupt Enable Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**CRC_BLKD_IE**	**CRC_TABORT_IE**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**CRC_BLKD_IE**	**CRC DMA Block Transfer Done Interrupt Enable Control** Enable this bit will generate the CRC DMA Transfer Done interrupt signal while CRC_BLKD_IF bit (CRCDMAISR [1] CRC DMA Block Transfer Done Interrupt Flag) is set to 1. 0 = Interrupt generator Disabled when CRC DMA transfer done. 1 = Interrupt generator Enabled when CRC DMA transfer done.
[0]	**CRC_TABORT_IE**	**CRC DMA Read/Write Target Abort Interrupt Enable Control** Enable this bit will generate the CRC DMA Target Abort interrupt signal while CRC_TARBOT_IF bit (CRCDMAISR [0] CRC DMA Read/Write Target Abort Interrupt Flag) is set to 1. 0 = Target abort interrupt generation Disabled during CRC DMA transfer. 1 = Target abort interrupt generation Enabled during CRC DMA transfer.

CRC DMA Interrupt Status Register (CRC_DMAISR)

Register	Offset	R/W	Description	Reset Value
CRC_DMAISR	CRC_BA+0x24	R/W	CRC DMA Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**CRC_BLKD_IF**	**CRC_TABORT_IF**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**CRC_BLKD_IF**	**CRC DMA Block Transfer Done Interrupt Flag** This bit indicates that CRC DMA transfer has finished or not. 0 = Not finished if TRIG_EN (CRC_CTL[23]) has enabled. 1 = CRC transfer done if TRIG_EN (CRC_CTL[23]) has enabled. (When CRC DMA transfer done, TRIG_EN (CRC_CTL[23]) will be cleared automatically) **Note:** This bit is cleared by writing "1" to it.
[0]	**CRC_TABORT_IF**	**CRC DMA Read/Write Target Abort Interrupt Flag** This bit indicates that CRC bus has error or not during CRC DMA transfer. 0 = No bus error response received during CRC DMA transfer. 1 = Bus error response received during CRC DMA transfer. **Note1:** This bit is cleared by writing "1" to it. **Note2:** This bit indicates bus master received error response or not. If bus master received error response, it means that CRC transfer target abort is happened. DMA will stop transfer and respond this event to user then CRC state machine goes to IDLE state. When target abort occurred, user must reset DMA before transfer those data again

.

CRC Write Data Register (CRC_WDATA)

Register	Offset	R/W	Description	Reset Value
CRC_WDATA	CRC_BA+0x80	R/W	CRC Write Data Register	0x0000_0000

31	30	29	28	27	26	25	24
CRC_WDATA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CRC_WDATA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CRC_WDATA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRC_WDATA							

Bits	Description	
[31:0]	**CRC_WDATA**	**CRC Write Data Bits** When operating in CPU mode, user can write data to this field to perform CRC operation. When operating in DMA mode, this field indicates the DMA read data from memory and cannot be written. **Note:** When the write data length is 8-bit mode, the valid data in CRC_WDATA register is only CRC_WDATA [7:0] bits; if the write data length is 16-bit mode, the valid data in CRC_WDATA register is only CRC_WDATA [15:0].

CRC Seed Register (CRC_SEED)

Register	Offset	R/W	Description	Reset Value
CRC_SEED	CRC_BA+0x84	R/W	CRC Seed Register	0xFFFF_FFFF

31	30	29	28	27	26	25	24
CRC_SEED							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CRC_SEED							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CRC_SEED							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRC_SEED							

Bits	Description	
[31:0]	**CRC_SEED**	**CRC Seed Value** This field indicates the CRC seed value.

CRC Checksum Register (CRC_CHECKSUM)

Register	Offset	R/W	Description	Reset Value
CRC_CHECKSUM	CRC_BA+0x88	R	CRC Checksum Register	0xFFFF_FFFF

31	30	29	28	27	26	25	24
CRC_CHECKSUM							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CRC_CHECKSUM							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CRC_CHECKSUM							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRC_CHECKSUM							

Bits	Description	
[31:0]	**CRC_CHECKSUM**	**CRC Checksum Results** This fields indicates the CRC checksum result

DMA Global Control Register (DMA_GCRCSR)

Register	Offset	R/W	Description	Reset Value
DMA_GCRCSR	DMA_GCR_BA+0x00	R/W	DMA Global Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							CRC_CLK_EN
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			CLK4_EN	CLK3_EN	CLK2_EN	CLK1_EN	Reserved
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24]	**CRC_CLK_EN**	**CRC Controller Clock Enable Control** 0 = Disabled. 1 = Enabled.
[23:13]	**Reserved**	Reserved.
[12]	**CLK4_EN**	**PDMA Controller Channel 4 Clock Enable Control** 0 = Disabled. 1 = Enabled.
[11]	**CLK3_EN**	**PDMA Controller Channel 3 Clock Enable Control** 0 = Disabled. 1 = Enabled.
[10]	**CLK2_EN**	**PDMA Controller Channel 2 Clock Enable Control** 0 = Disabled. 1 = Enabled.
[9]	**CLK1_EN**	**PDMA Controller Channel 1 Clock Enable Control** 0 = Disabled. 1 = Enabled.
[8:0]	**Reserved**	Reserved.

DMA Service Selection Control Register 0 (DMA_DSSR0)

Register	Offset	R/W	Description	Reset Value
DMA_DSSR0	DMA_GCR_BA+0x04	R/W	DMA Service Selection Control Register 0	0x1F1F_1F00

31	30	29	28	27	26	25	24
Reserved			**CH3_SEL**				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			**CH2_SEL**				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**CH1_SEL**				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28:24]	**CH3_SEL**	**Channel 3 Selection** This filed defines which peripheral is connected to PDMA channel 3. User can configure the peripheral setting by CH3_SEL. The channel configuration is the same as CH1_SEL field. Please refer to the explanation of CH1_SEL.
[23:21]	**Reserved**	Reserved.
[20:16]	**CH2_SEL**	**Channel 2 Selection** This filed defines which peripheral is connected to PDMA channel 2. User can configure the peripheral setting by CH2_SEL. The channel configuration is the same as CH1_SEL field. Please refer to the explanation of CH1_SEL.
[15:13]	**Reserved**	Reserved.
[12:8]	**CH1_SEL**	**Channel 1 Selection** This filed defines which peripheral is connected to PDMA channel 1. User can configure the peripheral by setting CH1_SEL. 00000 = Connect to SPI0_TX. 00001 = Connect to SPI1_TX. 00010 = Connect to UART0_TX. 00011 = Connect to UART1_TX. 00100 = Reserved. 00101 = Reserved. 00110 = Reserved. 00111 = Reserved. 01000 = Reserved. 01001 = Connect to TMR0. 01010 = Connect to TMR1. 01011 = Connect to TMR2. 01100 = Connect to TMR3. 10000 = Connect to SPI0_RX.

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

Bits	Description	
		10001 = Connect to SPI1_RX. 10010 = Connect to UART0_RX. 10011 = Connect to UART1_RX. 10100 = Reserved. 10101 = Reserved. 10110 = Connect to ADC. 10111 = Reserved. 11000 = Reserved. 11001 = Connect to PWM0_CH0. 11010 = Connect to PWM0_CH2. 11011 = Reserved. 11100 = Reserved.. Others = Disable to connected any peripheral.
[7:0]	**Reserved**	Reserved.

DMA Service Selection Control Register 1 (DMA_DSSR1)

Register	Offset	R/W	Description	Reset Value
DMA_DSSR1	DMA_GCR_BA+0x08	R/W	DMA Service Selection Control Register 1	0x0000_001F

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			CH4_SEL				

Bits	Description	
[31:5]	Reserved	Reserved.
[4:0]	CH4_SEL	**Channel 4 Selection** This filed defines which peripheral is connected to PDMA channel 4. User can configure the peripheral by setting CH4_SEL. 00000 = Connect to SPI0_TX. 00001 = Connect to SPI1_TX. 00010 = Connect to UART0_TX. 00011 = Connect to UART1_TX. 00100 = Reserved. 00101 = Reserved. 00110 = Reserved. 00111 = Reserved. 01000 = Reserved. 01001 = Connect to TMR0. 01010 = Connect to TMR1. 01011 = Connect to TMR2. 01100 = Connect to TMR3. 10000 = Connect to SPI0_RX. 10001 = Connect to SPI1_RX. 10010 = Connect to UART0_RX. 10011 = Connect to UART1_RX. 10100 = Reserved. 10101 = Reserved. 10110 = Connect to ADC. 10111 = Reserved. 11000 = Reserved. 11001 = Connect to PWM0_CH0. 11010 = Connect to PWM0_CH2. 11011 = Reserved.

Bits	Description	
		11100 = Reserved. Others = Disable to connected any peripheral.

DMA Global Interrupt Status Register (DMA_GCRISR)

Register	Offset	R/W	Description	Reset Value
DMA_GCRISR	DMA_GCR_BA+0x0C	R	DMA Global Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**INTRCRC**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**INTR4**	**INTR3**	**INTR2**	**INTR1**	**Reserved**

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**INTRCRC**	**Interrupt Status of CRC Controller (Read Only)** This bit is the interrupt status of CRC controller
[15:5]	**Reserved**	Reserved.
[4]	**INTR4**	**Interrupt Status of Channel 4 (Read Only)** This bit is the interrupt status of PDMA channel4.
[3]	**INTR3**	**Interrupt Status of Channel 3 (Read Only)** This bit is the interrupt status of PDMA channel3.
[2]	**INTR2**	**Interrupt Status of Channel 2 (Read Only)** This bit is the interrupt status of PDMA channel2. **Note:** This bit is read only
[1]	**INTR1**	**Interrupt Status of Channel 1 (Read Only)** This bit is the interrupt status of PDMA channel1.
[0]	**Reserved**	Reserved.

6.9 Timer Controller

6.9.1 Overview

This chip is equipped with four timer modules including TIMER0, TIMER1, TIMER2 and TIMER3, which allow user to easily implement a counting scheme or timing control for applications. The timer can perform functions like frequency measurement, event counting, interval measurement, clock generation, delay timing, and so on. The timer can generate an interrupt signal upon timeout, or provide the current value of count during operation.

6.9.2 Features

- Independent Clock Source for each Timer (TMRx_CLK, x= 0, 1,2,3)
- Time-out period = (Period of timer clock input) * (8-bit pre-scale counter + 1) * (24-bit TCMP)
- Counting cycle time = (1 / TMRx_CLK) * (2^8) * (2^{24})
- Internal 8-bit pre-scale counter
- Internal 24-bit up counter is readable through TDR (Timer Data Register)
- Supports One-shot, Periodic, Output Toggle and Continuous Counting Operation mode
- Supports external pin capture for interval measurement
- Supports external pin capture for timer counter reset
- Supports Inter-Timer trigger
- Supports event generator in TIMER 0 and TIMER 2 to generate event to TIMER1 and TIMER3, respectively.
- Supports Internal trigger event to ADC and PDMA

6.9.3 Block Diagram

Each timer is equipped with an 8-bit pre-scale counter, a 24-bit up-counter, a 24-bit compare register and an interrupt request signal. Refer to Figure 6‑28 And Figure 6‑29 For the timer controller block diagram. There are five options of clock sources for each timer, Figure 6‑30 Illustrate the Clock Source control function.

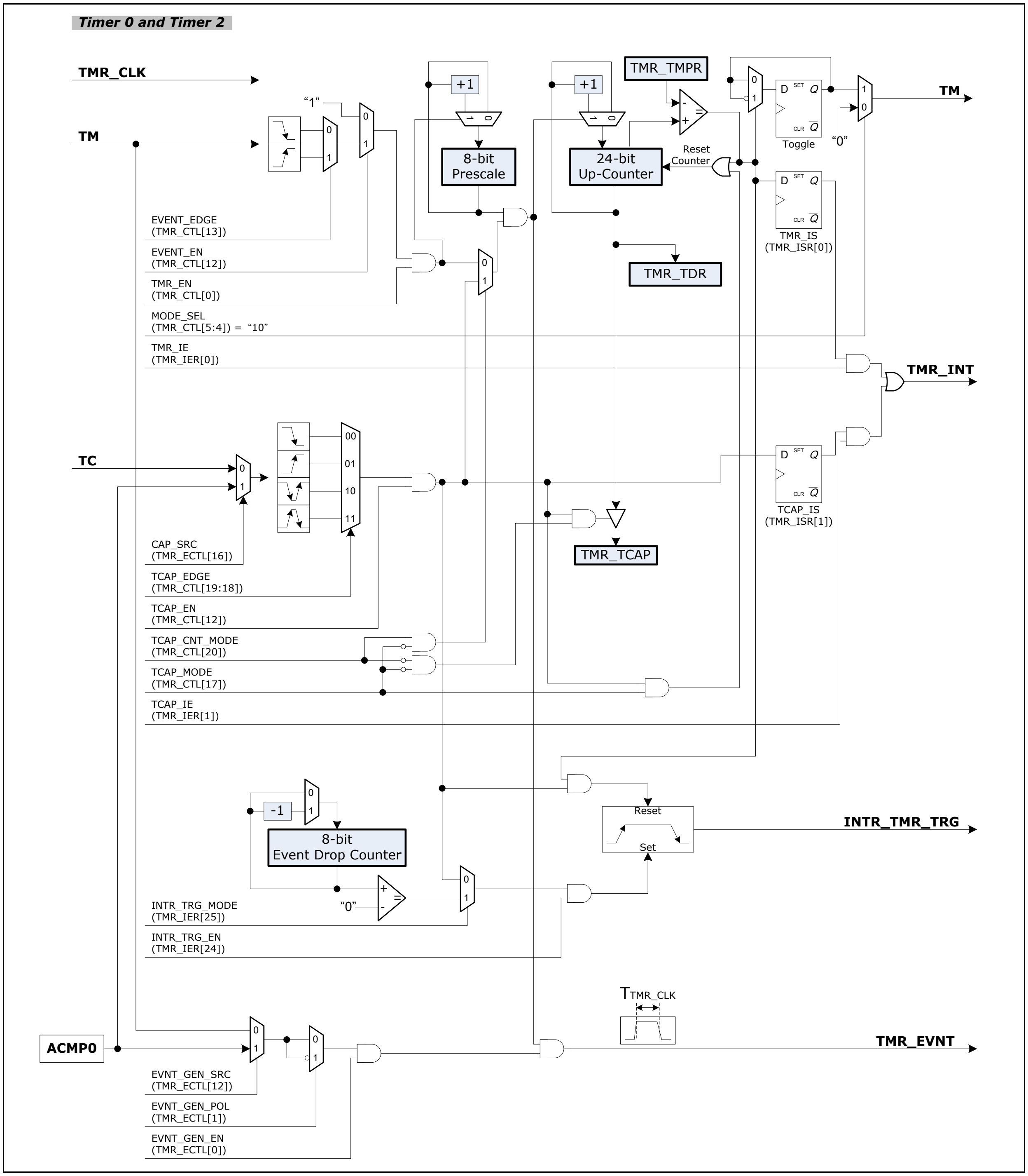


Figure 6‑28 Timer 0 and Timer 2 Controller Block Diagram

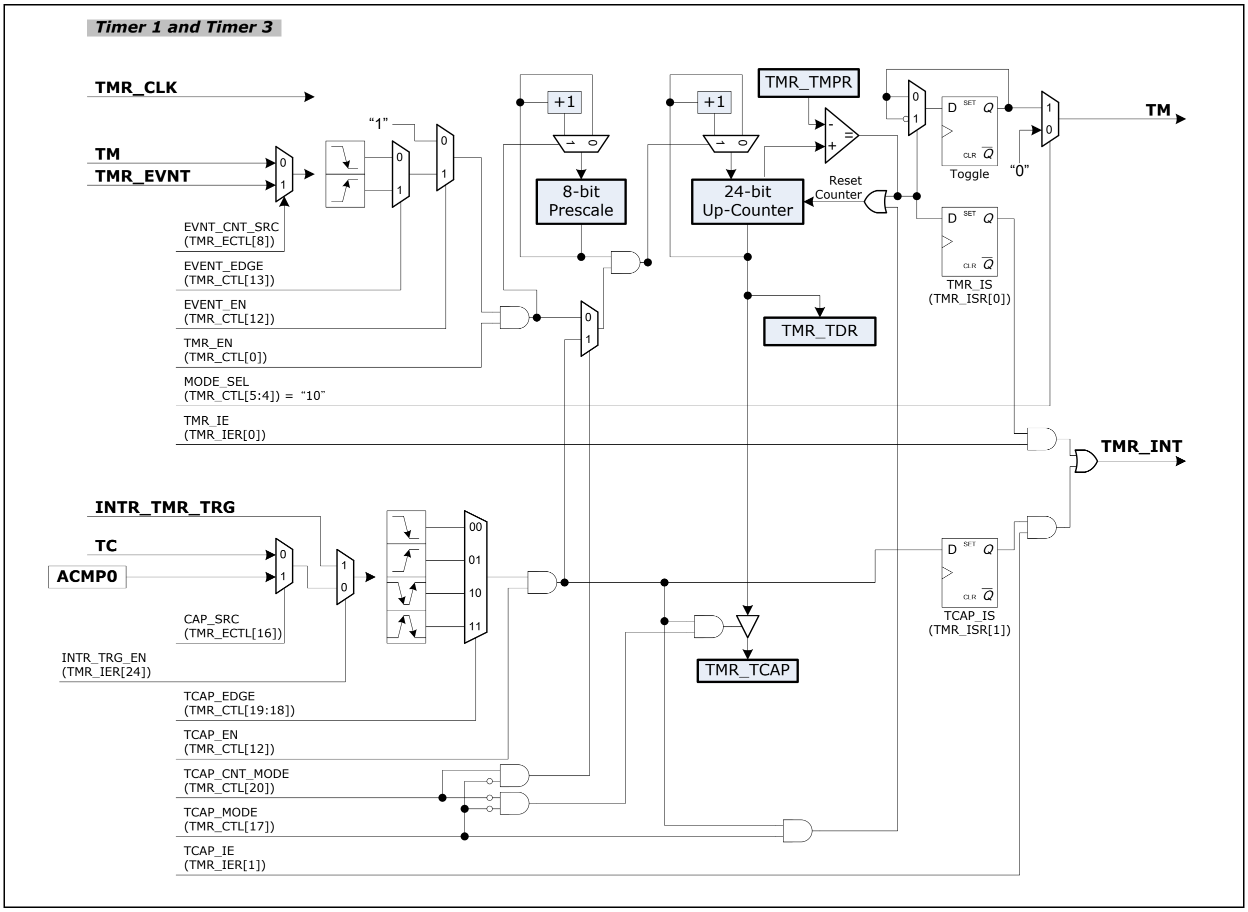
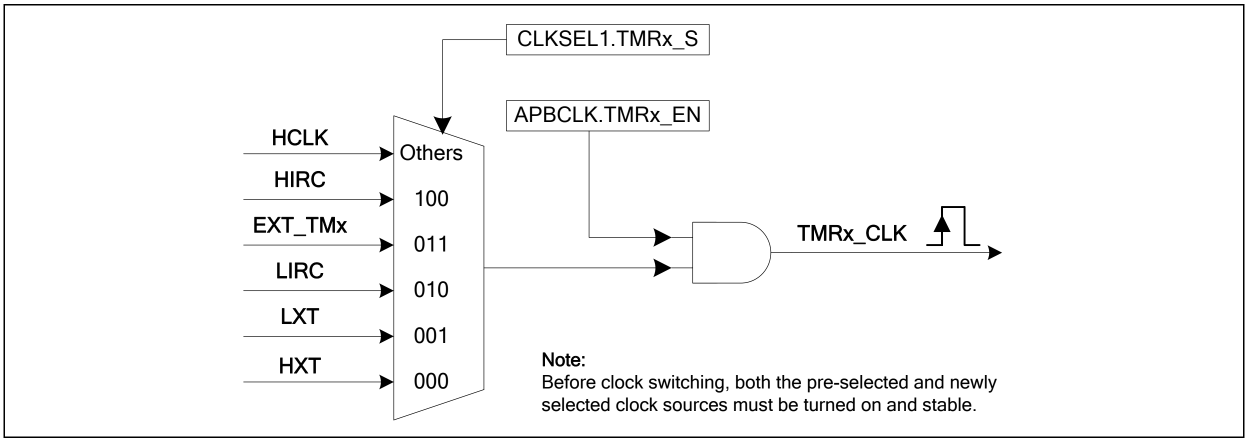


Figure 6‑29 Timer 1 and Timer 3 Controller Block Diagram



Note:
Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.

Figure 6‑30 Timer Clock Controller Diagram

6.9.4 Functional Description

Timer controller provides One-shot, Period, Toggle and Continuous Counting operation modes. The event counting function is also provided to count the events/counts from external pin and external pin capture function for interval measurement or reset timer counter. In addition, timer controller provides the Inter-Timer Trigger Mode to measure input frequency precisely. Each operating function mode is shown as follows:

6.9.4.1 One-Shot Mode

If the timer is operated in One-shot mode (MODE_SEL (TMRx_CTL[5:4]) is 00) and TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 1, the timer counter starts up counting. Once the timer counter value (TMRx_DR) reaches timer compare register (TMRx_CMPR) value, the TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1. If TMR_IE (TMRx_IER[0] timer interrupt enable bit) is set to 1, and TMR_IS (TMRx_ISR[0] timer interrupt status) is 1 then the interrupt signal is generated and sent to NVIC to inform CPU for indicating that the timer counting overflow happens. If TMR_IE (TMRx_IER[0] timer interrupt enable bit) is set to 0, no interrupt signal is generated.

In this operating mode, once the timer counter value (TMRx_DR) reaches timer compare register (TMRx_CMPR) value, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1, timer counting operation stops and the timer counter value (TMRx_DR) goes back to counting initial value then TMR_EN (TMRx_CTL[0] timer counter enable bit) is cleared to 0 by timer controller automatically. That is to say, timer operates timer counting and compares with TMRx_CMPR value function only one time after programming the timer compare register (TMRx_CMPR) value and TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 1. So, this operating mode is called One-Shot mode.

6.9.4.2 Periodic Mode

If the timer is operated in Period mode (MODE_SEL (TMRx_CTL[5:4]) is 01) and TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 1, the timer counter starts up counting. Once the timer counter value (TMRx_DR) reaches timer compare register (TMRx_CMPR) value, the TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1. If TMR_IE (TMRx_IER[0] timer interrupt enable bit) is set to 1, and TMR_IS (TMRx_ISR[0] timer interrupt status) is 1 then the interrupt signal is generated and sent to NVIC to inform CPU for indicating that the timer counting overflow happens. If TMR_IE (TMRx_IER[0] timer interrupt enable bit) is set to 0, no interrupt signal is generated.

In this operating mode, once the timer counter value (TMRx_DR) reaches timer compare register (TMRx_CMPR) value, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1, the timer counter value (TMRx_DR) goes back to counting initial value and TMR_EN (TMRx_CTL[0] timer counter enable bit) is kept at 1 (counting enable continuously) and timer counter operates up counting again. If TMR_IS (TMRx_ISR[0] timer interrupt status) is cleared by software, once the timer counter value (TMRx_DR) reaches timer compare register (TMRx_CMPR) value again, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1 also. That is to say, timer operates timer counting and compares with TMRx_CMPR value function periodically. The timer counting operation does not stop until the TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 0. The interrupt signal is also generated periodically. So, this operating mode is called Periodic mode.

6.9.4.3 Toggle Mode

If the timer is operated in Toggle mode (MODE_SEL (TMRx_CTL[5:4]) is 10) and TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 1, the timer counter starts up counting. Once the timer counter value (TMRx_DR) reaches timer compare register (TMRx_CMPR) value, the TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1. If TMR_IE (TMRx_IER[0] timer interrupt enable bit) is set to 1, and TMR_IS (TMRx_ISR[0] timer interrupt status) is 1 then the interrupt signal is generated and sent to NVIC to inform CPU for indicating that the timer counting overflow happens. If TMR_IE

(TMRx_IER[0] timer interrupt enable bit) is set to 0, no interrupt signal is generated.

In this operating mode, once the timer counter value (TMRx_DR) reaches timer compare register (TMRx_CMPR) value, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1, toggle out signal is inverted, the timer counter value (TMRx_DR) goes back to counting initial value and TMR_EN (TMRx_CTL[0] timer counter enable bit) is kept at 1 (counting enable continuously) and timer counter operates up counting again. If TMR_IS (TMRx_ISR[0] timer interrupt status) is cleared by software, once the timer counter value (TMRx_DR) reaches timer compare register (TMRx_CMPR) value again, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1 also and toggle out signal is inverted again. The timer counting operation does not stop until the TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 0. Thus, the toggle output signal is changing back and forth with 50% duty cycle. So, this operating mode is called Toggle mode.

6.9.4.4 Continuous Counting Mode

If the timer is operated in Continuous Counting mode (MODE_SEL (TMRx_CTL[5:4]) is 11) and TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 1, the timer counter starts up counting. Once the timer counter value (TMRx_DR) reaches timer compare register (TMRx_CMPR) value, the TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1. If TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 1, and TMR_IS (TMRx_ISR[0] timer interrupt status) is 1 then the interrupt signal is generated and sent to NVIC to inform CPU for indicating that the timer counting overflow happens. If TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 0, no interrupt signal is generated.

In this operating mode, once the timer counter value (TMRx_DR) reaches timer compare register (TMRx_CMPR) value, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1 and TMR_EN (TMRx_CTL[0] timer counter enable bit) is kept at 1 (counting enable continuously) and timer counter continuous counting without reload the timer counter value (TMRx_DR) to counting initial value. User can change different timer compare register (TMRx_CMPR) value immediately without disabling timer counter and restarting timer counter counting.

For example, the timer compare register (TMRx_CMPR) value is set as 80, first. (The timer compare register (TMRx_CMPR) should be less than 2^24 and be greater than 1). Once the timer counter value (TMRx_DR) reaches to 80, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1 and TMR_EN (TMRx_CTL[0] timer counter enable bit) is kept at 1 (counting enable continuously) and timer counter value (TMRx_DR) will not goes back to 0, it continues counting to 81, 82, 83,⋯ to (2^24 -1) then 0, 1, 2, 3, ⋯ to 2^24 -1 again and again. Next, if user programs timer compare register (TMRx_CMPR) value as 200 and the TMR_IS (TMRx_ISR[0] timer interrupt status) is cleared to 0, then TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1 again when timer counter value (TMRx_DR) reaches to 200. At last, user programs timer compare register (TMRx_CMPR) value as 500 and clears TMR_IS (TMRx_ISR[0] timer interrupt status) to 0, then TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1 again when timer counter value (TMRx_DR) reaches to 500. In this mode, the timer counter value (TMRx_DR) is keeping up counting always even if TMR_IS (TMRx_ISR[0] timer interrupt status) is 1. So, this operation mode is called as Continuous Counting mode.

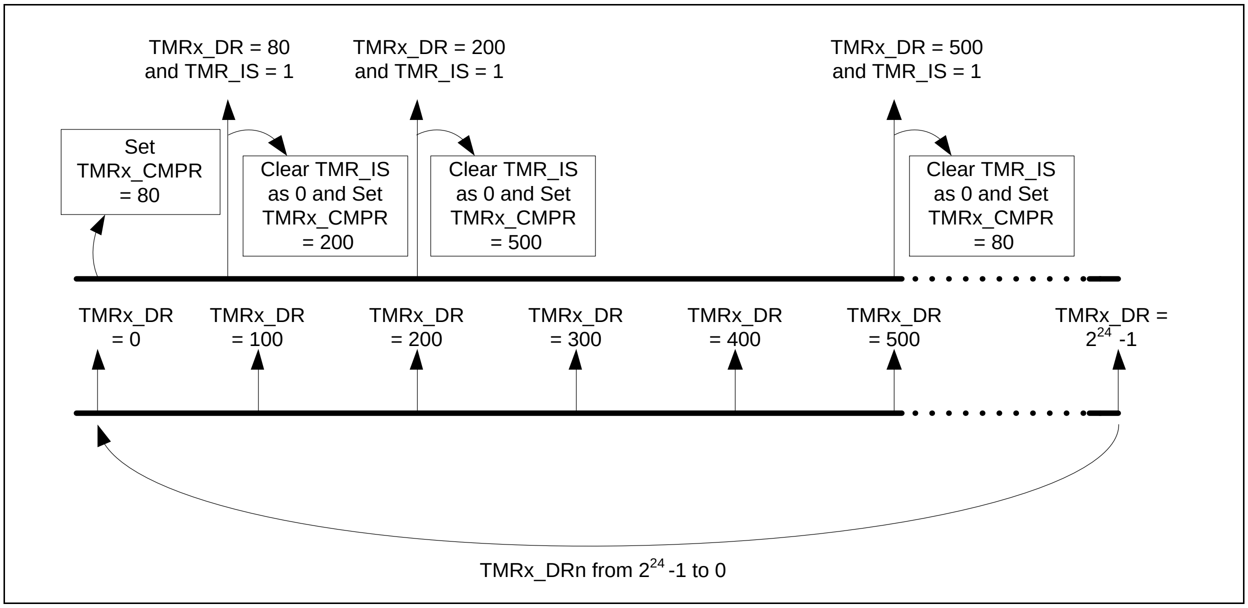


Figure 6‑31 Continuous Counting Mode

6.9.4.5 Event Counting Mode

An application which can count the events/counts from external event input pin is called as event counting function. In this mode, most of the timer control registers are the same with the timer operating function mode except EVENT_EN (TMRx_CTL[12] event counting mode enable) has to set to 1. When status transition on external event input pin, the event counter value (TMRx_DR value) will be counted according to EVENT_EDGE (TMRx_CTL[13] event counting mode edge selection) setting. EVNT_DEB_EN (TMRx_CTL[14] external event de-bounce enable) bit is for enabled or disabled edge detection de-bounce circuit of external event input pin. The max frequency of event counting source on external event input pin should be less than 1/4 TMRx_CLK if EVNT_DEB_EN (TMRx_CTL[14] external event de-bounce enable) is 0 or less than 1/10 TMRx_CLK if EVNT_DEB_EN (TMRx_CTL[14] external event de-bounce enable) is 1. Otherwise, the event counter value (TMRx_DR value) will not be counted normally.

6.9.4.6 Timer Counter Capture/Reset Function

In this mode, Timer will monitor the transition of external pin to save the 24-bit counter value or reset the 24-bit counter.

If TCAP_MODE (TMRx_CTL[17]) is 0, the transition on external pin is used as timer counter capture function. In this mode, if TCAP_CNT_MOD (TMRx_CTL[20]) is 0, the free-counting mode, 24-bit up-counting timer will keep counting continuously. And when the transition of external pin matches the TCAP_EDGE (TMRx_CTL[19:18]) setting, the value of 24-bit up-counting timer will be saved into register TMRx_TCAP. If TCAP_CNT_MOD (TMRx_CTL[20]) is 1, the trigger-counting mode, 24-bit up-counting timer will keep its value at 0. Once the transition of external pin matches the 1st transition of TCAP_EDGE (TMRx_CTL[19:18]) setting, the 24-bit up-counting timer will start counting. And then if the transition of external pin matches the 2nd transition of TCAP_EDGE (TMRx_CTL[19:18]) setting, the 24-bit up-counting timer value will be saved into register TMRx_TCAP. At the same time, if MODE_SEL (TMRx_CTL[5:4]) is 00, the 24-bit up-counting timer stops the counting; if MODE_SEL (TMRx_CTL[5:4]) is 01 or 10 and TCAP_EDGE (TMRx_CTL[19:18]) is 10 or 11, the 24-bit up-counting timer reset and restart the counter from 0; if MODE_SEL (TMRx_CTL[5:4]) is 11, the 24-bit up-counting timer continues the counting.

If TCAP_MODE (TMRx_CTL[17]) is 1, the transition on external pin is used as timer counter reset function. In this mode, once the transition of external pin matches the TCAP_EDGE (TMRx_CTL[19:18]) setting, the 24-bit up-counting timer will be reset.

To detect the transition of external pin, the timer circuit implements the de-bounce circuit for external pin. Based on the result of de-bounce circuit and external pin, the rising-edge or falling-edge could be detected. The reset value of de-bounce circuit is “0” and the de-bounce would only active when both TCAP_DEB_EN (TMRx_CTL[22]) and TCAP_EN (TMRx_CTL[16]) are enabled. So, if the external pin level is “1” and TCAP_EDGE (TMRx_CTL[19:18]) is set to detect rising-edge of external pin, then after de-bounce circuit active (TCAP_DEB_EN (TMRx_CTL[22]) is “1” and TCAP_EN (TMRx_CTL[16]) is “1”), a false rising-edge would be detected. This would result in the incorrect capture data (TMRx_TCAP) while 1st time the TCAP_IS (TMRx_ISR[1]) is set. To avoid this incorrect capture data to affect the capture application, discard this 1st capture data is necessary and recommended.

6.9.4.7 Inter-Timer Trigger Mode

In this mode, the TMRx (where x=0 or 2), will be forced in counter mode, counting with external event, and will generate a signal (INTR_TMR_TRG) to trigger TMRx+1 (where x=0 or 2). The TMRx+1 will be forced in trigger-counting mode of capture function.

While INTR_TRG_EN (TMRx_CTL[24], x=0 or 2) is set, the TMRx will make a rising-edge transition of trigger signal (INTR_TMR_TRG) to TMRx+1 while 24-bit counter is counting from 0 to 1. And when 24-bit counter reaches the 24-bit TCMPR value, the TMRx will make a falling-edge transition of trigger signal (INTR_TMR_TRG) to TMRx+1.

When INTR_TMR_TRG transited from low to high (rising-edge), the 24-bit counter of TMRx+1 will start to count. Also, when INTR_TMR_TRG transited from high to low (falling-edge), the 24-bit counter of TMRx+1 will stop counting. At the same time, the value of 24-bit counter will be saved into TMRx+1_TCAP.

The example shown below described how inter-timer trigger mode work. When inter-timer trigger mode is enabled (INTR_TRG_EN (TMRx_CTL[24]) = 1), the TMRx_Counter starts to up-counting in each external event (TMRx) detected. When TRMx_Counter counted from 0 to 1, the INTR_TMR_TRG is set high, and TMRx+1_Counter is then start counting in each TMRx+1_CLK. When TMRx_Counter reaches the value of TMRx_CMPR, it stops counting and ITNR_TMR_TRG is set low. And then, the TMRx+1_Counter stops counting, too. In the same time, the value of TMRx+1_Counter is sampled into TMRx+1_TCAP and interrupt status TMR_IS (TMRx_ISR[0]) is also set. In addition, the INTR_TRG_EN (TRMx_CTL[24]) is also automatically cleared by H/W.

By using Inter-timer Trigger mode, the period of external event (TMRx) could be measured more precisely. For the example shown below when TMRx received 100 events, the counter of TMRx+1 counting to 999. If TMRx+1_CLK frequency is 10 MHz, then we know the time for 100 events is about 99900ns. Therefore, the period of an external event is 999ns and the frequency of external event will be 1.001 MHz.

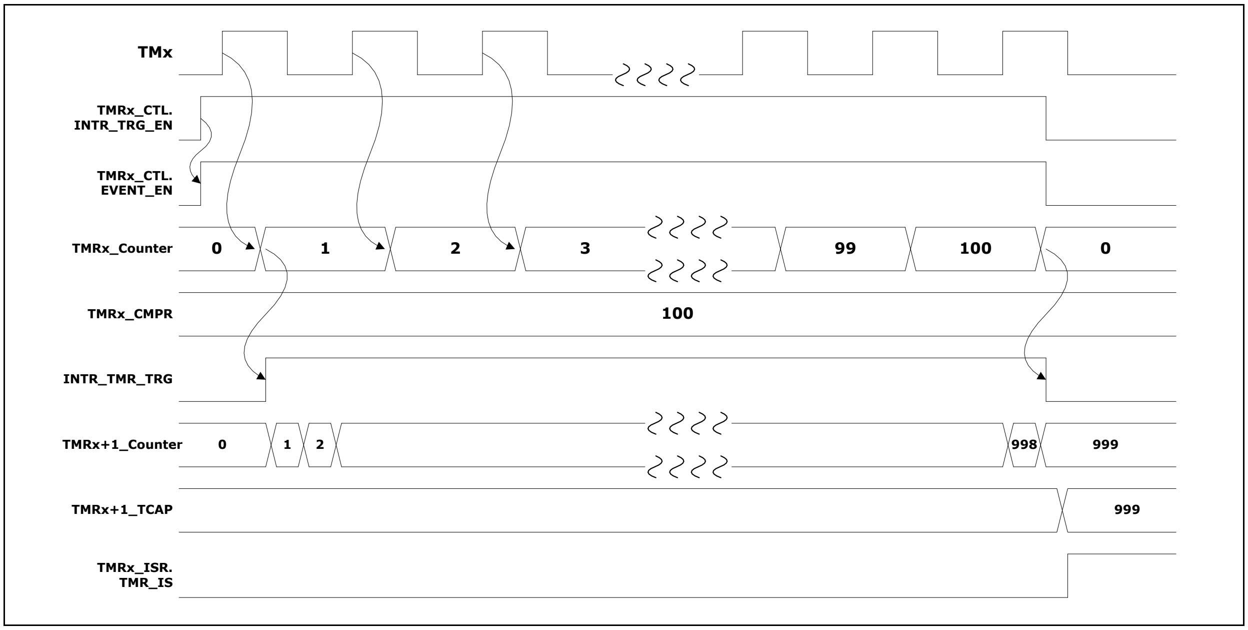


Figure 6-32 Inter-Timer Trigger Mode

6.9.4.8 Event Generator Mode

Timer 0 and Timer 2 equipped a function to monitor a signal and generate a pulse event out to Timer 1 and Timer 3 respectively. Timer 1 and Timer 3 use this internal pulse event as the external event from pin TM1 and TM3. In Timer 1 and Timer 3, they use EVNT_CNT_SRC (TMR_ECTL[8]) to select the event source is from external event pin (TM1 and TM3) or from internal pulse event generated by Event Generator Mode of Timer 0 and Timer 2.

When EVNT_GEN_EN (TMRx_ECTL[0]) is high, timer starts to monitor the polarity of signal defined by EVNT_GEN_SRC (TMRx_ECTL[12]). If the polarity of signal defined by EVNT_GEN_SRC (TMRx_ECTL[12]) is the same as the polarity define by EVNT_GEN_POL (TMRx_ECTL[1]), timer would generate a high pulse event out when it increase the 24-bit up counter. The width of this high pulse is a clock cycle of TMRx_CLK.

For example, if the EVNT_GEN_EN (TMRx_ECTL[0]) and EVNT_GEN_POL (TMRx_ECTL[1]) are both high, timer monitor the polarity of signal defined by EVNT_GEN_SRC (TMRx_ECTL[12]). If the polarity of signal defined by EVNT_GEN_SRC (TMRx_ECTL[12]) is high, timer generate a high pulse out (pulse width is a clock cycle of TRMx_CLK). If the polarity of signal defined by EVNT_GEN_SRC (TMRx_ECTL[12]) is low, timer keep the out signal in low.

The table below described the event generator related settings and result.

EVNT_GEN_EN (TMR_ECTL[0])	EVNT_GEN_POL (TMR_ECTL[1])	Event Generator Reference Input Source Signal	Event Generator Output
0	X	X	0
1	0	0	High Pulse
1	0	1	0
1	1	0	0
1	1	1	High Pulse

6.9.5 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
TMR Base Address: **TMR0_BA = 0x4001_0000** **TMR1_BA = 0x4001_0100** **TMR2_BA = 0x4011_0000** **TMR3_BA = 0x4011_0100**				
TMR0_CTL	TMR0_BA+0x000	R/W	Timer 0 Control Register	0x0000_0000
TMR0_PRECNT	TMR0_BA+0x004	R/W	Timer 0 Pre-Scale Counter Register	0x0000_0000
TMR0_CMPR	TMR0_BA+0x008	R/W	Timer 0 Compare Register	0x0000_0000
TMR0_IER	TMR0_BA+0x00C	R/W	Timer 0 Interrupt Enable Register	0x0000_0000
TMR0_ISR	TMR0_BA+0x010	R/W	Timer 0 Interrupt Status Register	0x0000_0000
TMR0_DR	TMR0_BA+0x014	R/W	Timer 0 Data Register	0x0000_0000
TMR0_TCAP	TMR0_BA+0x018	R	Timer 0 Capture Data Register	0x0000_0000
TMR0_ECTL	TMR0_BA+0x020	R/W	Timer 0 Extended Control Register	0x0000_0002
TMR1_CTL	TMR1_BA+0x000	R/W	Timer 1 Control Register	0x0000_0000
TMR1_PRECNT	TMR1_BA+0x004	R/W	Timer 1 Pre-Scale Counter Register	0x0000_0000
TMR1_CMPR	TMR1_BA+0x008	R/W	Timer 1 Compare Register	0x0000_0000
TMR1_IER	TMR1_BA+0x00C	R/W	Timer 1 Interrupt Enable Register	0x0000_0000
TMR1_ISR	TMR1_BA+0x010	R/W	Timer 1 Interrupt Status Register	0x0000_0000
TMR1_DR	TMR1_BA+0x014	R/W	Timer 1 Data Register	0x0000_0000
TMR1_TCAP	TMR1_BA+0x018	R	Timer 1 Capture Data Register	0x0000_0000
TMR1_ECTL	TMR1_BA+0x020	R/W	Timer 1 Extended Control Register	0x0000_0002
TMR2_CTL	TMR2_BA+0x000	R/W	Timer 2 Control Register	0x0000_0000
TMR2_PRECNT	TMR2_BA+0x004	R/W	Timer 2 Pre-Scale Counter Register	0x0000_0000
TMR2_CMPR	TMR2_BA+0x008	R/W	Timer 2 Compare Register	0x0000_0000
TMR2_IER	TMR2_BA+0x00C	R/W	Timer 2 Interrupt Enable Register	0x0000_0000
TMR2_ISR	TMR2_BA+0x010	R/W	Timer 2 Interrupt Status Register	0x0000_0000
TMR2_DR	TMR2_BA+0x014	R/W	Timer 2 Data Register	0x0000_0000
TMR2_TCAP	TMR2_BA+0x018	R	Timer 2 Capture Data Register	0x0000_0000
TMR2_ECTL	TMR2_BA+0x020	R/W	Timer 2 Extended Control Register	0x0000_0002

TMR3_CTL	TMR3_BA+0x000	R/W	Timer 3 Control Register	0x0000_0000
TMR3_PRECNT	TMR3_BA+0x004	R/W	Timer 3 Pre-Scale Counter Register	0x0000_0000
TMR3_CMPR	TMR3_BA+0x008	R/W	Timer 3 Compare Register	0x0000_0000
TMR3_IER	TMR3_BA+0x00C	R/W	Timer 3 Interrupt Enable Register	0x0000_0000
TMR3_ISR	TMR3_BA+0x010	R/W	Timer 3 Interrupt Status Register	0x0000_0000
TMR3_DR	TMR3_BA+0x014	R/W	Timer 3 Data Register	0x0000_0000
TMR3_TCAP	TMR3_BA+0x018	R	Timer 3 Capture Data Register	0x0000_0000
TMR3_ECTL	TMR3_BA+0x020	R/W	Timer 3 Extended Control Register	0x0000_0002
GPA_SHADOW	TMR0_BA+0x200	R	GPIO Port A Pin Value Shadow Register	0x0000_XXXX
GPB_SHADOW	TMR0_BA+0x204	R	GPIO Port B Pin Value Shadow Register	0x0000_XXXX
GPC_SHADOW	TMR0_BA+0x208	R	GPIO Port C Pin Value Shadow Register	0x0000_XXXX
GPD_SHADOW	TMR0_BA+0x20C	R	GPIO Port D Pin Value Shadow Register	0x0000_XXXX
GPE_SHADOW	TMR0_BA+0x210	R	GPIO Port E Pin Value Shadow Register	0x0000_XXXX
GPF_SHADOW	TMR0_BA+0x214	R	GPIO Port F Pin Value Shadow Register	0x0000_XXXX

6.9.6 Register Description

Timer Control Register (TMR_CTL)

Register	Offset	R/W	Description	Reset Value
TMR0_CTL	TMR0_BA+0x000	R/W	Timer 0 Control Register	0x0000_0000
TMR1_CTL	TMR1_BA+0x000	R/W	Timer 1 Control Register	0x0000_0000
TMR2_CTL	TMR2_BA+0x000	R/W	Timer 2 Control Register	0x0000_0000
TMR3_CTL	TMR3_BA+0x000	R/W	Timer 3 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						INTR_TRG_MODE	INTR_TRG_EN
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	TCAP_DEB_EN	Reserved	TCAP_CNT_MOD	TCAP_EDGE		TCAP_MODE	TCAP_EN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	EVNT_DEB_EN	EVENT_EDGE	EVENT_EN	TMR_TRG_SEL	PDMA_TEEN	Reserved	ADC_TEEN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TMR_ACT	ACMP_EN_TMR	MODE_SEL		DBGACK_EN	WAKE_EN	SW_RST	TMR_EN

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25]	**INTR_TRG_MODE**	**Inter-timer Trigger Mode Selection** This bit controls the timer operation mode when inter-timer trigger function is enabled. When this bit is low, the TMRx will be in counter mode and counting with external Clock Source or event. In addition, TMRx+1 will be in trigger-counting mode of capture function. In this mode, TMRx_CMPR control when inter-timer trigger function terminated. When this bit is high, the TMRx will be in counter mode and counting with external Clock Source or event. In addition, TMRx+1 will be in trigger-counting mode of capture function. In this mode, TMRx+1_CMPR control when inter-timer trigger function terminated. In this mode, TMRx would ignore some incoming event based on the EVNT_DROP_CNT (TMRx_ECTL[31:24]). And once the TMRx+1 counter value equal or large than TMRx+1_CMPR, TMRx would terminate the operation when next incoming event received. 0 = Inter-Timer Trigger function wouldn't ignore any incoming event. 1 = Inter-Timer Trigger function would ignore incoming event based on the EVNT_DROP_CNT (TMRx_ECTL[31:24]). **Note:** For TMRx+1_CTL, this bit is ignored and the read back value is always 0.

Bits	Description	
[24]	**INTR_TRG_EN**	**Inter-timer Trigger Function Enable Control** This bit controls if Inter-timer Trigger function is enabled. If Inter-timer Trigger function is enabled, the TMRx will be in counter mode and counting with external Clock Source or event. In addition, TMRx+1 will be in trigger-counting mode of capture function. 0 = Inter-timer trigger function Disabled. 1 = Inter-timer trigger function Enabled. **Note:** For TMRx+1_CTL, this bit is ignored and the read back value is always 0.
[23]	**Reserved**	Reserved.
[22]	**TCAP_DEB_EN**	**TC Pin De-bounce Enable Control** When CAP_DEB_EN (TMRx_CTL[22]) is set, the TC pin de-bounce circuit will be enabled to eliminate the bouncing of the signal. In de-bounce circuit the TC pin signal will be sampled 4 times by TMRx_CLK. 0 = De-bounce circuit Disabled. 1 = De-bounce circuit Enabled. **Note:** When TCAP_EN (TMRx_CTL[16]) is enabled, enable this bit is recommended. And, while TCAP_EN (TMRx_CTL[16]) is disabled, disable this bit is recommended to save power consumption. **Note:** When CAP_SRC (TMRx_ECTL[16]) is high, the capture signal is from internal of chip and the de-bounce circuit would not take effect no matter this bit is high or low. **Note:** For Timer 1 and 3, when INTR_TRG_EN (TMRx_CTL[24]) is high, the capture signal is from internal of chip and the de-bounce circuit would not take effect no matter this bit is high or low.
[21]	**Reserved**	Reserved.
[20]	**TCAP_CNT_MOD**	**Timer Capture Counting Mode Selection** This bit indicates the behavior of 24-bit up-counting timer while TCAP_EN (TMRx_CTL[16]) is set to high. If this bit is 0, the free-counting mode, the behavior of 24-bit up-counting timer is defined by MODE_SEL (TMRx_CTL[5:4]) field. When TCAP_EN (TMRx_CTL[16]) is set, TCAP_MODE (TMRx_CTL[17]) is 0, and the transition of TC pin matches the TCAP_EDGE (TMRx_CTL[19:18]) setting, the value of 24-bit up-counting timer will be saved into register TMRx_TCAP. If this bit is 1, Trigger-counting mode, 24-bit up-counting timer will be not counting and keep its value at 0. When TCAP_EN (TMRx_CTL[16]) is set, TCAP_MODE (TMRx_CTL[17]) is 0, and once the transition of external pin matches the 1st transition of TCAP_EDGE (TMRx_CTL[19:18]) setting, the 24-bit up-counting timer will start counting. And then if the transition of external pin matches the 2nd transition of TCAP_EDGE (TMRx_CTL[19:18]) setting, the 24-bit up-counting timer will stop counting. And its value will be saved into register TMRx_TCAP. 0 = Capture with free-counting timer mode. 1 = Capture with trigger-counting timer mode. **Note:** For TMRx+1_CTL, if INTR_TRG_EN (TMRx_CTL[24]) is set, the TCAP_CNT_MOD will be forced to high, the capture with Trigger-counting Timer mode (where x = 0 or 2).

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[19:18]</td><td><b>TCAP_EDGE</b></td><td><b>TC Pin Edge Detect Selection</b>
This field defines that active transition of TC pin is for timer counter reset function or for timer capture function.
For timer counter reset function and free-counting mode of timer capture function, the configurations are:
<table>
<tr><th>TCAP_EDGE</th><th>Output Clock (MCLK)</th></tr>
<tr><td>00</td><td>A falling edge (1 to 0 transition) on TC pin is an active transition.</td></tr>
<tr><td>01</td><td>A rising edge (0 to 1 transition) on TC pin is an active transition.</td></tr>
<tr><td>10</td><td>Both falling edge (1 to 0 transition) and rising edge (0 to 1 transition) on TC pin are active transitions.</td></tr>
<tr><td>11</td><td>Both falling edge (1 to 0 transition) and rising edge (0 to 1 transition) on TC pin are active transitions.</td></tr>
</table>
For trigger-counting mode of timer capture function, the configurations are:
<table>
<tr><th>TCAP_EDGE</th><th>Output clock (MCLK)</th></tr>
<tr><td>00</td><td>1st falling edge on TC pin triggers 24-bit timer to start counting, while 2nd falling edge triggers 24-bit timer to stop counting.</td></tr>
<tr><td>01</td><td>1st rising edge on TC pin triggers 24-bit timer to start counting, while 2nd rising edge triggers 24-bit timer to stop counting.</td></tr>
<tr><td>10</td><td>Falling edge on TC pin triggers 24-bit timer to start counting, while rising edge triggers 24-bit timer to stop counting.</td></tr>
<tr><td>11</td><td>Rising edge on TC pin triggers 24-bit timer to start counting, while falling edge triggers 24-bit timer to stop counting.</td></tr>
</table>
<b>Note:</b> For TMRx+1_CTL, if INTR_TRG_EN (TMRx_CTL[24]) is set, the TCAP_EDGE will be forced to 11 (where x = 0 or 2).</td></tr>
<tr><td>[17]</td><td><b>TCAP_MODE</b></td><td><b>TC Pin Function Mode Selection</b>
This bit indicates if the transition on TC pin is used as timer counter reset function or timer capture function.
0 = The transition on TC pin is used as timer capture function.
1 = The transition on TC pin is used as timer counter reset function.
<b>Note:</b> For TMRx+1_CTL, if INTR_TRG_EN (TMRx_CTL[24]) is set, the TCAP_MODE will be forced to low (where x = 0 or 2).</td></tr>
<tr><td>[16]</td><td><b>TCAP_EN</b></td><td><b>TC Pin Functional Enable Control</b>
This bit controls if the transition on TC pin could be used as timer counter reset function or timer capture function.
0 = The transition on TC pin is ignored.
1 = The transition on TC pin will result in the capture or reset of 24-bit timer counter.
<b>Note:</b> For TMRx_CTL, if INTR_TRG_EN (TMRx_CTL[24]) is set, the TCAP_EN will be forced to low and the TC pin transition is ignored (where x = 0 or 2).
<b>Note:</b> For TMRx+1_CTL, if INTR_TRG_EN (TMRx_CTL[24]) is set, the TCAP_EN will be forced to high (where x = 0 or 2).</td></tr>
<tr><td>[15]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
</table>

Bits	Description	
[14]	**EVNT_DEB_EN**	**External Event De-bounce Enable Control** When EVNT_DEB_EN is set, the external event pin de-bounce circuit will be enabled to eliminate the bouncing of the signal. In de-bounce circuit the external event pin will be sampled 4 times by TMRx_CLK. 0 = De-bounce circuit Disabled. 1 = De-bounce circuit Enabled. **Note:** When EVENT_EN (TMRx_CTL[12]) is enabled, enable this bit is recommended. And, while EVENT_EN (TMRx_CTL[12]) is disabled, disable this bit is recommended to save power consumption.
[13]	**EVENT_EDGE**	**Event Counting Mode Edge Selection** This bit indicates which edge of external event pin enabling the timer to increase 1. 0 = A falling edge of external event enabling the timer to increase 1. 1 = A rising edge of external event enabling the timer to increase 1.
[12]	**EVENT_EN**	**Event Counting Mode Enable Control** When EVENT_EN is set, the increase of 24-bit up-counting timer is controlled by external event pin. While the transition of external event pin matches the definition of EVENT_EDGE (TMRx_CTL[13]), the 24-bit up-counting timer increases by 1. Or, the 24-bit up-counting timer will keep its value unchanged. 0 = Timer counting is not controlled by external event pin. 1 = Timer counting is controlled by external event pin.
[11]	**TMR_TRG_SEL**	**Timer Trigger Selection** This bit controls if the TMR_IS (TMRx_ISR[0]) or TCAP_IS (TMRx_ISR[1]) is used to trigger PDMA and ADC while TMR_IS (TMRx_ISR[0]) or TCAP_IS (TMRx_ISR[1]) is set. If this bit is low and TMR_IS (TMRx_ISR[0]) is set, timer will generate an internal trigger event to PDMA or ADC while related trigger enable bit (PDMA_TEEN (TMRx_CTL[10]) or ADC_TEEN (TMRx_CTL[8])) is set. If this bit is set high and TCAP_IS (TMRx_ISR[0]) is set, timer will generate an internal trigger event to PDMA or ADC while related trigger enable bit (PDMA_TEEN (TMRx_CTL[10]) or ADC_TEEN (TMRx_CTL[8])) is set. 0 = TMR_IS (TMRx_ISR[0]) is used to trigger PDMA and ADC. 1 = TCAP_IS (TMRx_ISR[1]) is used to trigger PDMA and ADC.
[10]	**PDMA_TEEN**	**Timer Trigger PDMA Enable Control** This bit controls if TMR_IS (TMRx_ISR[0]) or TCAP_IS (TMRx_ISR[1]) could trigger PDMA. When PDMA_TEEN is set, TMR_IS (TMRx_ISR[0]) is set and the CAP_TRG_EN (TMRx_CTL[11]) is low, the timer controller will generate an internal trigger event to PDMA controller. When PDMA_TEEN is set, TCAP_IS (TMRx_ISR[1]) is set and the CAP_TRG_EN (TMRx_CTL[11]) is high, the timer controller will generate an internal trigger event to PDMA controller. 0 = TMR_IS (TMRx_ISR[0]) or TCAP_IS (TMRx_ISR[1]) trigger PDMA Disabled. 1 = TMR_IS (TMRx_ISR[0]) or TCAP_IS (TMRx_ISR[1]) trigger PDMA Enabled.
[9]	**Reserved**	Reserved.

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[8]</td><td><b>ADC_TEEN</b></td><td><b>Timer Trigger ADC Enable Control</b>
This bit controls if TMR_IS (TMRx_ISR[0]) or TCAP_IS (TMRx_ISR[1]) could trigger ADC.
When ADC_TEEN is set, TMR_IS (TMRx_ISR[0]) is set and the CAP_TRG_EN (TMRx_CTL[11]) is low, the timer controller will generate an internal trigger event to ADC controller.
When ADC_TEEN is set, TCAP_IS (TMRx_ISR[1]) is set and the CAP_TRG_EN (TMRx_CTL[11]) is high, the timer controller will generate an internal trigger event to ADC controller.
0 = TMR_IS (TMRx_ISR[0]) or TCAP_IS (TMRx_ISR[1]) trigger ADC Disabled.
1 = TMR_IS (TMRx_ISR[0]) or TCAP_IS (TMRx_ISR[1]) trigger ADC Enabled.</td></tr>
<tr><td>[7]</td><td><b>TMR_ACT</b></td><td><b>Timer Active Status Bit (Read Only)</b>
This bit indicates the timer counter status of timer.
0 = Timer is not active.
1 = Timer is in active.</td></tr>
<tr><td>[6]</td><td><b>ACMP_EN_TMR</b></td><td><b>ACMP Trigger Timer Enable Control</b>
This bit high enables the functionality that when ACMP0 is in sigma-delta mode, it could enable Timer.
0 = ACMP0 trigger timer functionality disabled.
1 = ACMP0 trigger timer functionality enabled.</td></tr>
<tr><td>[5:4]</td><td><b>MODE_SEL</b></td><td><b>Timer Operating Mode Select</b>
<table>
<tr><th>MODE_SEL</th><th>Output clock (MCLK)</th></tr>
<tr><td>00</td><td><b>The timer is operating in the one-shot mode.</b>
In this mode, the associated interrupt signal is generated (if TMR_IE (TMR_IER [0]) is enabled) once the value of 24-bit up counter equals the TMRx_CMPR. And TMR_EN (TMR_CTL [0]) is automatically cleared by hardware.</td></tr>
<tr><td>01</td><td><b>The timer is operating in the periodic mode.</b>
In this mode, the associated interrupt signal is generated periodically (if TMR_IE (TMR_IER [0]) is enabled) while the value of 24-bit up counter equals the TMRx_CMPR. After that, the 24-bit counter will be reset and starts counting from 0 again.</td></tr>
<tr><td>10</td><td><b>The timer is operating in the periodic mode with output toggling.</b>
In this mode, the associated interrupt signal is generated periodically (if TMR_IE (TMR_IER [0]) is enabled) while the value of 24-bit up counter equals the TMR_CMPR. After that, the 24-bit counter will be reset and starts counting from 0 again.
At the same time, timer controller will also toggle the output pin TM to its inverse level (from low to high or from high to low).
<b>Note:</b> The default level of TM after reset is low.</td></tr>
<tr><td>11</td><td><b>The timer is operating in continuous counting mode.</b>
In this mode, the associated interrupt signal is generated when TMR_DR = TMR_CMPR (if TMR_IE (TMR_IER [0]) is enabled). However, the 24-bit up-counter counts continuously without reset.</td></tr>
</table></td></tr>
</table>

Bits	Description	
[3]	**DBGACK_EN**	**ICE Debug Mode Acknowledge Ineffective Enable Control** 0 = ICE debug mode acknowledgement effects TIMER counting and TIMER counter will be held while ICE debug mode acknowledged. 1 = ICE debug mode acknowledgement is ineffective and TIMER counter will keep going no matter ICE debug mode acknowledged or not.
[2]	**WAKE_EN**	**Wake-up Enable Control** When WAKE_EN is set and the TMR_IS (TMRx_ISR[0]) or TCAP_IS (TMRx_ISR[1]) is set, the timer controller will generate a wake-up trigger event to CPU. 0 = Wake-up trigger event Disabled. 1 = Wake-up trigger event Enabled.
[1]	**SW_RST**	**Software Reset** Set this bit will reset the timer counter, pre-scale counter and also force TMR_EN (TMRx_CTL [0]) to 0. 0 = No effect. 1 **=** Reset Timer's pre-scale counter, internal 24-bit up-counter and TMR_EN (TMRx_CTL [0]) bit. **Note:** This bit will be auto cleared and takes at least 3 TMRx_CLK clock cycles.
[0]	**TMR_EN**	**Timer Counter Enable Control** 0 = Stops/Suspends counting. 1 = Starts counting. **Note1:** Set TMR_EN to 1 enables 24-bit counter keeps up counting from the last stop counting value. **Note2:** This bit is auto-cleared by hardware in one-shot mode (MODE_SEL (TMRx_CTL[5:4]) = 00) once the value of 24-bit up counter equals the TMRx_CMPR.

Timer Pre-Scale Counter Register (TMR_PRECNT)

Register	Offset	R/W	Description	Reset Value
TMR0_PRECNT	TMR0_BA+0x004	R/W	Timer 0 Pre-Scale Counter Register	0x0000_0000
TMR1_PRECNT	TMR1_BA+0x004	R/W	Timer 1 Pre-Scale Counter Register	0x0000_0000
TMR2_PRECNT	TMR2_BA+0x004	R/W	Timer 2 Pre-Scale Counter Register	0x0000_0000
TMR3_PRECNT	TMR3_BA+0x004	R/W	Timer 3 Pre-Scale Counter Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRESCALE_CNT							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**PRESCALE_CNT**	**Pre-scale Counter** Clock input is divided by PRESCALE_CNT + 1 before it is fed to the counter. If PRESCALE_CNT =0, then there is no scaling.

Timer Compare Register (TMR_CMPR)

Register	Offset	R/W	Description	Reset Value
TMR0_CMPR	TMR0_BA+0x008	R/W	Timer 0 Compare Register	0x0000_0000
TMR1_CMPR	TMR1_BA+0x008	R/W	Timer 1 Compare Register	0x0000_0000
TMR2_CMPR	TMR2_BA+0x008	R/W	Timer 2 Compare Register	0x0000_0000
TMR3_CMPR	TMR3_BA+0x008	R/W	Timer 3 Compare Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TMR_CMP							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TMR_CMP							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TMR_CMP							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**TMR_CMP**	**Timer Compared Value** TMR_CMP is a 24-bit compared register. When the internal 24-bit up-counter counts and its value is equal to TMR_CMP value, a Timer Interrupt is requested if the timer interrupt is enabled with TMR_EN (TMRx_CTL [0]) is enabled. The TMR_CMP value defines the timer counting cycle time. Time-out period = (Period of timer clock input) * (8-bit PRESCALE_CNT + 1) * (24-bit TMR_CMP). **Note1:** Never write 0x0 or 0x1 in TMR_CMP, or the core will run into unknown state. **Note2:** No matter TMR_EN (TMRx_CTL [0]) is 0 or 1, whenever software write a new value into this register, TIMER will restart counting using this new value and abort previous count.

Timer Interrupt Enable Register (TMR_IER)

Register	Offset	R/W	Description	Reset Value
TMR0_IER	TMR0_BA+0x00C	R/W	Timer 0 Interrupt Enable Register	0x0000_0000
TMR1_IER	TMR1_BA+0x00C	R/W	Timer 1 Interrupt Enable Register	0x0000_0000
TMR2_IER	TMR2_BA+0x00C	R/W	Timer 2 Interrupt Enable Register	0x0000_0000
TMR3_IER	TMR3_BA+0x00C	R/W	Timer 3 Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						TCAP_IE	TMR_IE

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**TCAP_IE**	**Timer Capture Function Interrupt Enable Control** 0 = Timer External Pin Function Interrupt Disabled. 1 = Timer External Pin Function Interrupt Enabled. **Note:** If timer external pin function interrupt is enabled, the timer asserts its interrupt signal when the TCAP_EN (TMRx_CTL[16]) is set and the transition of external pin matches the TCAP_EDGE (TMRx_CTL[19:18]) setting
[0]	**TMR_IE**	**Timer Interrupt Enable Control** 0 = Timer Interrupt Disabled. 1 = Timer Interrupt Enabled. **Note:** If timer interrupt is enabled, the timer asserts its interrupt signal when the associated counter is equal to TMR_CMPR.

Timer Interrupt Status Register (TMR_ISR)

Register	Offset	R/W	Description	Reset Value
TMR0_ISR	TMR0_BA+0x010	R/W	Timer 0 Interrupt Status Register	0x0000_0000
TMR1_ISR	TMR1_BA+0x010	R/W	Timer 1 Interrupt Status Register	0x0000_0000
TMR2_ISR	TMR2_BA+0x010	R/W	Timer 2 Interrupt Status Register	0x0000_0000
TMR3_ISR	TMR3_BA+0x010	R/W	Timer 3 Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	TCAP_IS_FEDGE	NCAP_DET_STS	TMR_WAKE_STS	Reserved		TCAP_IS	TMR_IS

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**TCAP_IS_FEDGE**	**TC Pin Edge Detect Is Falling** This flag indicates the edge detected in TC pin is rising edge or falling edge. Timer only updates this flag when it udpates the Timer Capture Data (TMR_TCAP[23:0]) value. When a new incoming capture event detected before CPU clearing the TCAP_IS (TMRx_ISR[1]) status, Timer will keep this bit unchanged. 0 = TC pin edge detected is rising edge. 1 = TC pin edge detected is falling edge.
[5]	**NCAP_DET_STS**	**New Capture Detected Status** This status is to indicate there is a new incoming capture event detected before CPU clearing the TCAP_IS (TMRx_ISR[1]) status. If the above condition occurred, the Timer will keep register TMRx_TCAP unchanged and drop the new capture value. Write 1 to clear this bit to 0. 0 = New incoming capture event didn't detect before CPU clearing TCAP_IS (TMRx_ISR[1]) status. 1 = New incoming capture event detected before CPU clearing TCAP_IS (TMRx_ISR[1]) status.
[4]	**TMR_WAKE_STS**	**Timer Wake-up Status** If timer causes CPU wakes up from Power-down mode, this bit will be set to high. It must be cleared by software with a write 1 to this bit. 0 = Timer does not cause system wake-up. 1 = Wakes system up from Power-down mode by Timer timeout.

Bits	Description	
[3:2]	**Reserved**	Reserved.
[1]	**TCAP_IS**	**Timer Capture Function Interrupt Status** This bit indicates the external pin function interrupt status of Timer. This bit is set by hardware when TCAP_EN (TMRx_CTL[16]) is set high, and the transition of external pin matches the TCAP_EDGE (TMRx_CTL[19:18]) setting. Write 1 to clear this bit to 0. If this bit is active and TCAP_IE (TMRx_IER[1]) is enabled, Timer will trigger an interrupt to CPU.
[0]	**TMR_IS**	**Timer Interrupt Status** This bit indicates the interrupt status of Timer. This bit is set by hardware when the up counting value of internal 24-bit counter matches the timer compared value (TMR_CMPR). Write 1 to clear this bit to 0. If this bit is active and TMR_IE (TMRx_IER[0]) is enabled, Timer will trigger an interrupt to CPU.

Timer Data Register (TMR_DR)

Register	Offset	R/W	Description	Reset Value
TMR0_DR	TMR0_BA+0x014	R/W	Timer 0 Data Register	0x0000_0000
TMR1_DR	TMR1_BA+0x014	R/W	Timer 1 Data Register	0x0000_0000
TMR2_DR	TMR2_BA+0x014	R/W	Timer 2 Data Register	0x0000_0000
TMR3_DR	TMR3_BA+0x014	R/W	Timer 3 Data Register	0x0000_0000

31	30	29	28	27	26	25	24
RSTACT	**Reserved**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TDR							

Bits	Description	
[31]	**RSTACT**	**Reset Active** This bit indicates if the counter reset operation active. When user write this register, timer starts to reset its internal 24-bit timer up-counter and 8-bit pre-scale counter to 0. In the same time, timer set this flag to 1 to indicate the counter reset operation is in progress. Once the counter reset operation done, timer clear this bit to 0 automatically. 0 = Reset operation done. 1 = Reset operation triggered by writing TMR_DR is in progress. **Note:** This bit is read only. Write operation wouldn't take any effect.
[30:24]	**Reserved**	Reserved.
[23:0]	**TDR**	**Timer Data Register (Read)** User can read this register for internal 24-bit timer up-counter value. **Counter Reset (Write)** User can write any value to this register to reset internal 24-bit timer up-counter and 8-bit pre-scale counter. This reset operation wouldn't affect any other timer control registers and circuit. After reset completed, the 24-bit timer up-counter and 8-bit pre-scale counter restart the counting based on the TMRx_CTL register setting.

Timer Capture Data Register (TMR_TCAP)

Register	Offset	R/W	Description	Reset Value
TMR0_TCAP	TMR0_BA+0x018	R	Timer 0 Capture Data Register	0x0000_0000
TMR1_TCAP	TMR1_BA+0x018	R	Timer 1 Capture Data Register	0x0000_0000
TMR2_TCAP	TMR2_BA+0x018	R	Timer 2 Capture Data Register	0x0000_0000
TMR3_TCAP	TMR3_BA+0x018	R	Timer 3 Capture Data Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
CAP							
15	14	13	12	11	10	9	8
CAP							
7	6	5	4	3	2	1	0
CAP							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**CAP**	**Timer Capture Data Register** When TCAP_EN (TMRx_CTL[16]) is set, TCAP_MODE (TMRx_CTL[17]) is 0, TCAP_CNT_MOD (TMRx_CTL[20]) is 0, and the transition of external pin matches the TCAP_EDGE (TMRx_CTL[19:18]) setting, the value of 24-bit up-counting timer will be saved into register TMRx_TCAP. When TCAP_EN (TMRx_CTL[16]) is set, TCAP_MODE (TMRx_CTL[17]) is 0, TCAP_CNT_MOD (TMRx_CTL[20]) is 1, and the transition of external pin matches the 2nd transition of TCAP_EDGE (TMRx_CTL[19:18]) setting, the value of 24-bit up-counting timer will be saved into register TMRx_TCAP. User can read this register to get the counter value. When a new incoming capture event detected before CPU clearing the TCAP_IS (TMRxISR[1]) status, Timer will keep this filed value unchanged and drop the new capture value.

Timer Extended Control Register (TMR_ECTL)

Register	Offset	R/W	Description	Reset Value
TMR0_ECTL	TMR0_BA+0x020	R/W	Timer 0 Extended Control Register	0x0000_0002
TMR1_ECTL	TMR1_BA+0x020	R/W	Timer 1 Extended Control Register	0x0000_0002
TMR2_ECTL	TMR2_BA+0x020	R/W	Timer 2 Extended Control Register	0x0000_0002
TMR3_ECTL	TMR3_BA+0x020	R/W	Timer 3 Extended Control Register	0x0000_0002

31	30	29	28	27	26	25	24
EVNT_DROP_CNT							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							CAP_SRC
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			EVNT_GEN_SRC	Reserved			EVNT_CNT_SRC
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						EVNT_GEN_POL	EVNT_GEN_EN

Bits	Description	
[31:24]	**EVNT_DROP_CNT**	**Event Drop Count** This field indicates timer to drop how many events after inter-timer trigger function enable. For example, if user writes 0x7 to this field, timer would drop 7 first incoming events and starts the inter-timer trigger operation when it get 8th event. This field would affect timer's operation only when inter-timer trigger function enabled (INTR_TRG_EN (TMRx_CTL[24]) = 1) and ITNR_TRG_MODE (TMRx_CTL[25]) = 1.
[23:17]	**Reserved**	Reserved.
[16]	**CAP_SRC**	**Capture Function Source Selection** This bit defines timer counter reset function or timer capture function controlled by transition of TC pin or transition of internal signals from other functional blocks of this chip. 0 = Transition of TC pin selected. 1 = Transition of internal signals from ACMP0. **Note:** When this bit is high, the EVNT_DEB_EN (TMRx_CTL[14]) would not take effect.
[15:13]	**Reserved**	Reserved.
[12]	**EVNT_GEN_SRC**	**Event Generator Reference Input Source Selection** This bit defines the event generator function controlled by external event pin or internal event signals from ACMP0. 0 = The event generator reference source is from external event pin. 1 = The event generator reference source is from ACMP0. **Note:** This bit is only available in TMRx (where x = 0 or 2).
[11:9]	**Reserved**	Reserved.

Bits	Description	
[8]	**EVNT_CNT_SRC**	**Event Counting Source Selection** This bit defines the TMRx+1 event counting source is from external event pin TMx+1 or internal signal from TMRx's event generator output (where x = 0 or 2). 0 = The event counting source is from external event pin. 1 = The event counting source is from TMRx's event generator output. **Note:** This bit is only available in TMRx+1 (where x = 0 or 2).
[7:2]	**Reserved**	Reserved.
[1]	**EVNT_GEN_POL**	**Event Generator Reference Input Source Polarity Selection** When this bit is low and EVNT_GEN_EN (TMRx_ECTL[0]) is high, timer would generate a high pulse event out when it increases the 24-bit up counter and the polarity of signal defined by EVNT_GEN_SRC (TMRx_ECTL[12]) is low. When this bit is high and EVNT_GEN_EN (TMRx_ECTL[0]) is high, timer would generate a low pulse event pulse out when it increases the 24-bit up counter and the polarity of signal defined by EVNT_GEN_SRC (TMRx_ECTL[12]) is high. This bit only affects timer's operation when EVNT_GEN_EN (TMRx_ECTL[0]) is high. 0 = Timer generates a high pulse event out when it increase the 24-bit up counter and the polarity of signal defined by EVNT_GEN_SRC (TMRx_ECTL[12]) is low. 1 = Timer generates a high pulse event out when it increase the 24-bit up counter and the polarity of signal defined by EVNT_GEN_SRC (TMRx_ECTL[12]) is high.
[0]	**EVNT_GEN_EN**	**Event Generator Function Enable Control** When this bit is high, timer would generate a high pulse event out when it increases the 24-bit up counter and the polarity of signal defined by EVNT_GEN_SRC (TMRx_ECTL[12]) is same as the polarity defined by EVNT_GEN_POL (TMRx_ECTL[1]). 0 = Event generator function disabled. 1 = Event generator function enabled.

GPIO Port Pin Value Shadow Register (GPx_SHADOW, x = A, B, C, D, E, F)

Register	Offset	R/W	Description	Reset Value
GPA_SHADOW	TMR0_BA+0x200	R	GPIO Port A Pin Value Shadow Register	0x0000_XXXX
GPB_SHADOW	TMR0_BA+0x204	R	GPIO Port B Pin Value Shadow Register	0x0000_XXXX
GPC_SHADOW	TMR0_BA+0x208	R	GPIO Port C Pin Value Shadow Register	0x0000_XXXX
GPD_SHADOW	TMR0_BA+0x20C	R	GPIO Port D Pin Value Shadow Register	0x0000_XXXX
GPE_SHADOW	TMR0_BA+0x210	R	GPIO Port E Pin Value Shadow Register	0x0000_XXXX
GPF_SHADOW	TMR0_BA+0x214	R	GPIO Port F Pin Value Shadow Register	0x0000_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PIN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PIN							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[n] n = 0,1..15	**PIN**	**GPIO Port [A/B/C/D/E/F] Pin Values** The value read from each of these bit reflects the actual status of the respective GPIO pin. These registers are shadow registers of GPIOx_PIN register. **Note 1:** For GPE_SHADOW, bits [15:10] are reserved. **Note 1:** For GPF_SHADOW, bits [15:6] are reserved.

6.10 Pulse Width Modulation (PWM)

6.10.1 Overview

This chip has one PWM controller, which includes 4 independent PWM outputs, CH0~CH3, or as 2 complementary PWM pairs, (CH0, CH1), (CH2, CH3) with 2 programmable dead-zone generators.

Each of the two PWM outputs, (CH0, CH1), (CH2, CH3), share the same 8-bit prescaler, clock divider providing 5 divided frequencies (1, 1/2, 1/4, 1/8, 1/16). Each PWM output has independent 16-bit PWM counter which has two counting modes for PWM period control. The PWM counter operates as down counting in edge-aligned mode and up-down counting in center-aligned mode only. Each PWM output also has a 16-bit comparator for PWM duty control. Each dead-zone generator has two outputs. The first dead-zone generator output is CH0 and CH1, and for the second dead-zone generator, the output is CH2 and CH3. The PWM controller total provide four independent PWM interrupt flags which are set by hardware when the corresponding PWM period down counter in edge-aligned mode (or up-down counter in center-aligned mode) reaches 0. PWM interrupt will be asserted when both PWM interrupt source and its corresponding enable bit are active. Each PWM output can be configured as one-shot mode to produce only one PWM cycle signal or continuous mode to output PWM waveform continuously.

When DZEN01(PWM_CTL[4]) is set, CH0 and CH1 perform complementary PWM paired function; the paired PWM timing, period, duty and dead-time are determined by PWM channel 0 timer and Dead-zone generator 0. Similarly, When DZEN23(PWM_CTL[5]) is set the complementary PWM pair of (CH2, CH3) is controlled by PWM channel 2.

To prevent PWM driving output pin with unsteady waveform, the 16-bit period down counter and 16-bit comparator are implemented with double buffer. When user writes data to counter/comparator buffer registers the updated value will be loaded into the 16-bit down counter/ comparator at the time down counter reaching 0. The double buffering feature avoids glitch at PWM outputs.

When the 16-bit period down counter reaches 0, the interrupt request is generated. If PWM output is set as continuous mode, when the down counter reaches 0, it is reloaded with CN of PWM_DUTYy(y=0~3) Register automatically then start decreases, repeatedly. If the PWM output is set as one-shot mode, the down counter will stop and generate one interrupt request when it reaches 0.

The value of PWM counter comparator is used for pulse width modulation. The counter control logic changes the output level when down-counter value matches the value of compare register.

The alternate feature of the PWM is digital input capture function. If capture function is enabled the PWM output pin is switched as capture input pin. The capture channel 0 and PWM CH0 share one timer; and the capture channel 1 and PWM CH1 share one timer, and etc. Therefore user must set up the PWM timer before enabling capture feature. After capture feature of channel 0 is enabled, the capture always latches PWM CH0 timer value to Capture Rising Latch Register CRL (PWM_CRL0[15:0]) when input channel has a rising transition and latches PWM CH0 timer value to Capture Falling Latch Register CFL (PWM_CFL0[15:0]) when input channel has a falling transition. Capture channel 0 interrupt is programmable by setting CRL_IE0(PWM_CAPINTEN[0]) for rising transition or CFL_IE0 (PWM_CAPINTEN[1]) for falling transition. Whenever Capture rising event latched for channel 0, the PWM CH0 timer will be reload at this moment if the corresponding reload enable bit CAPRELOADREN0 (PWM_CAPCTL[6]) is set.

The maximum captured frequency that PWM can capture is dominated by the capture interrupt latency. When capture interrupt occurs, software will do at least three steps, they are: Read PWMINTSTS to tell it from interrupt source and Read PWM_CRLy/PWM_CFLy(y=0~3) to get capture value and finally write 1 to clear PWM_INTSTS. If interrupt latency will take time T0 to finish, the capture signal mustn’t transient during this interval. In this case, the maximum capture frequency will be 1/T0.

6.10.2 Features

6.10.2.1 PWM Function:

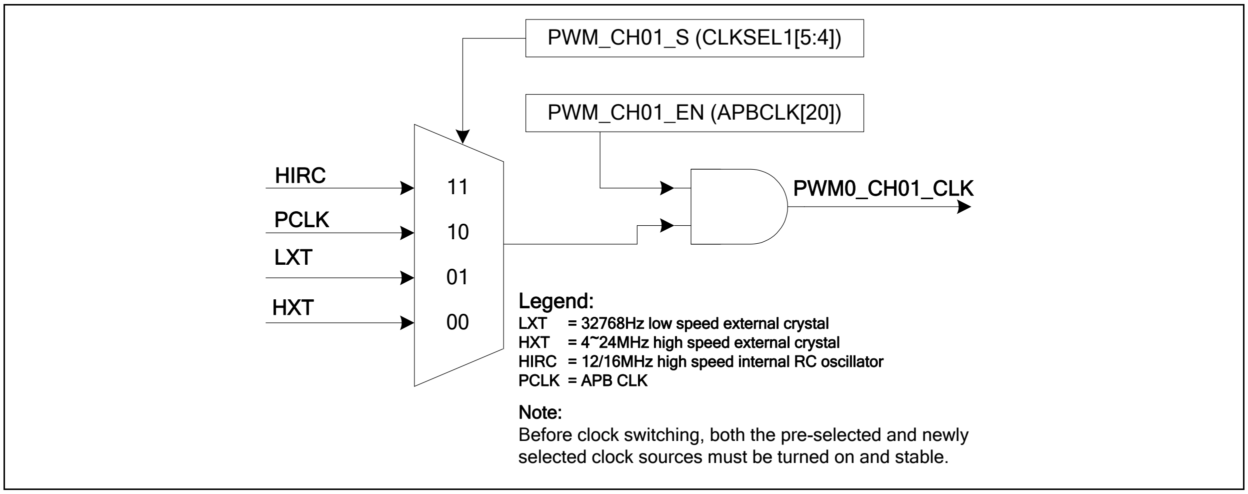
- PWM controllers has 4 independent PWM outputs, CH0~CH3, or as 2 complementary PWM pairs, (CH0, CH1), (CH2, CH3) with 2 programmable dead-zone generators
- Up to 4 PWM channels or 2 PWM paired channels
- Up to 16 bits PWM counter width
- PWM Interrupt request synchronous with PWM period
- Single-shot or Continuous mode
- Two Dead-Zone generators

6.10.2.2 Capture Function:

- Timing control logic shared with PWM timer.
- 4 Capture input channels shared with 4 PWM output channels.
- Each channel supports one rising latch register CRL (PWM_CRL0[15:0]), one falling latch register CFL (PWM_CFL0[15:0]) and Capture interrupt flag CAPIF0 (PWM_CAPINTSTS[0]).
- Four 16-bit counters for four capture channels or two 32-bit counter for two capture channels when cascade is enabled: when CH01CASKEN (PWM_CAPCTL[13]) is set, the original 16-bit counter of channel 1 will combine with channel 0’s 16 bit counter for channel 0 input capture counting and so does CH23CASKEN(PWM_CAPCTL[29]) for channel 2, 3
- Supports PDMA transfer function for PWM channel 0, 2

6.10.3 Block Diagram

The following figures illustrate the architecture of PWM in groups. (Timer 0&1 are in one group and timer 2&3 are in another, and so on.)



Legend:
LXT = 32768Hz low speed external crystal
HXT = 4~24MHz high speed external crystal
HIRC = 12/16MHz high speed internal RC oscillator
PCLK = APB CLK

Note:
Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.

Figure 6-33 PWM Clock for Channel 0, 1

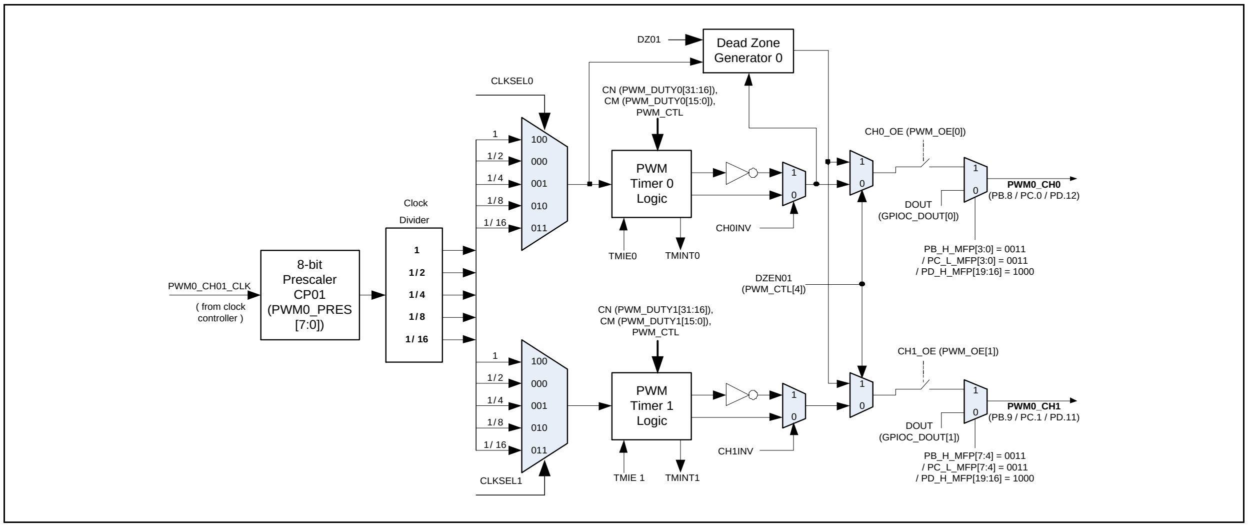


Figure 6-34 PWM Generator for Channel 0, 1

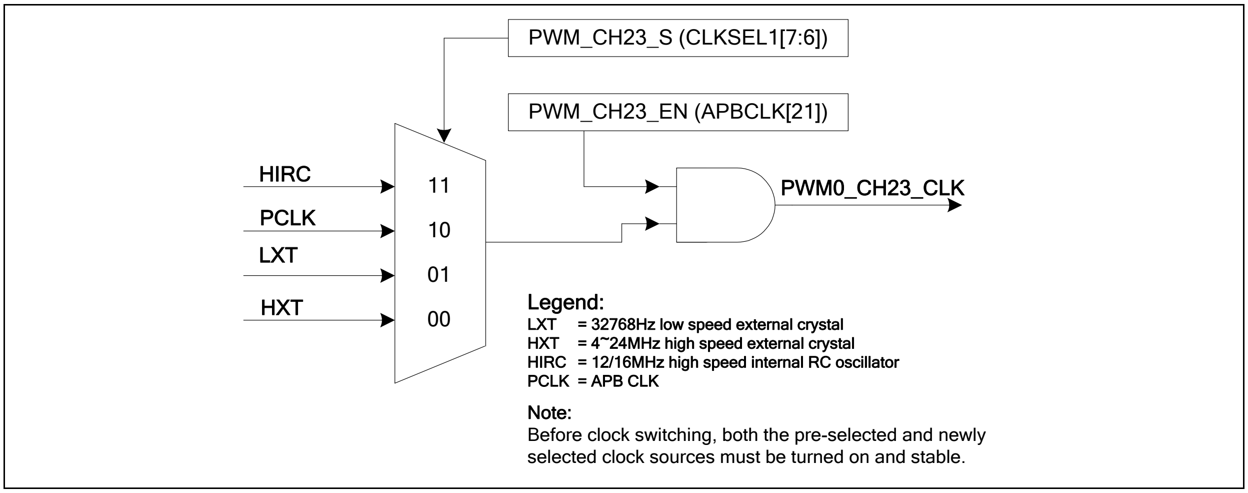

PWM_CH23_S (CLKSEL1[7:6])
PWM_CH23_EN (APBCLK[21])
HIRC
PCLK
LXT
HXT
11
10
01
00
PWM0_CH23_CLK
Legend:
LXT = 32768Hz low speed external crystal
HXT = 4~24MHz high speed external crystal
HIRC = 12/16MHz high speed internal RC oscillator
PCLK = APB CLK
Note:
Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.

Figure 6-35 PWM Clock for Channel 2, 3

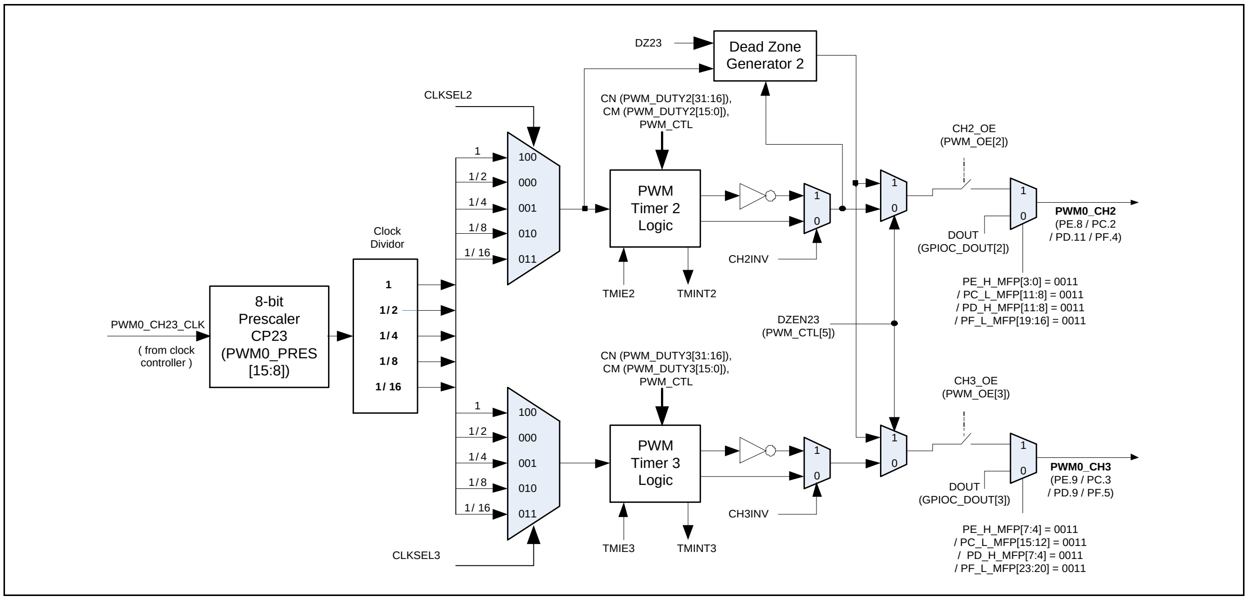

DZ23
Dead Zone Generator 2
CLKSEL2
CN (PWM_DUTY2[31:16]), CM (PWM_DUTY2[15:0]), PWM_CTL
1
1/2
1/4
1/8
1/16
100
000
001
010
011
PWM Timer 2 Logic
TMIE2
TMINT2
CH2INV
CH2_OE (PWM_OE[2])
DOUT (GPIOC_DOUT[2])
PWM0_CH2
(PE.8 / PC.2 / PD.11 / PF.4)
PE_H_MFP[3:0] = 0011 / PC_L_MFP[11:8] = 0011 / PD_H_MFP[11:8] = 0011 / PF_L_MFP[19:16] = 0011
Clock Dividor
PWM0_CH23_CLK
(from clock controller)
8-bit Prescaler CP23 (PWM0_PRES [15:8])
DZEN23 (PWM_CTL[5])
CN (PWM_DUTY3[31:16]), CM (PWM_DUTY3[15:0]), PWM_CTL
PWM Timer 3 Logic
CH3INV
CH3_OE (PWM_OE[3])
DOUT (GPIOC_DOUT[3])
PWM0_CH3
(PE.9 / PC.3 / PD.9 / PF.5)
PE_H_MFP[7:4] = 0011 / PC_L_MFP[15:12] = 0011 / PD_H_MFP[7:4] = 0011 / PF_L_MFP[23:20] = 0011
CLKSEL3
TMIE3
TMINT3

Figure 6-36 PWM Generator for Channel 2, 3

6.10.4 Basic Configuration

The PWM pin functions are configured in PB_H_MFP, PC_L_MFP, PD_H_MFP, PE_H_MFP, PF_L_MFP registers.

6.10.5 Functional Description

6.10.5.1 PWM-Timer Operation

The PWM period and duty control are decided by PWM_DUTYy (y=0~3) register CN (PWM_DUTYy[15:0]) and CM (PWM_DUTYy[31:16]). The PWM-timer supports two types operation: Edge-aligned type and Center-aligned type.

6.10.5.2 Edge-aligned Type

The PWM-timer timing operation in Edge-aligned type and the pulse width modulation following the formula below and the legend of PWM-Timer Comparator are shown in the following figures. Note that the corresponding I/O pins must be configured as output type before PWM function is enabled.

PWM frequency = PWMxy_CLK/(prescale+1)*(clock divider)/(CN+1); where xy, could be 01, 23, depending on selected PWM channel.

Duty ratio = (CM+1)/(CN+1).

CM >= CN: PWM output is always high.

CM < CN: PWM low width= (CN-CM) unit1; PWM high width = (CM+1) unit.

If CM = 0: PWM low width = (CN) unit; PWM high width = 1 unit

Note: 1. Unit = one PWM clock cycle.

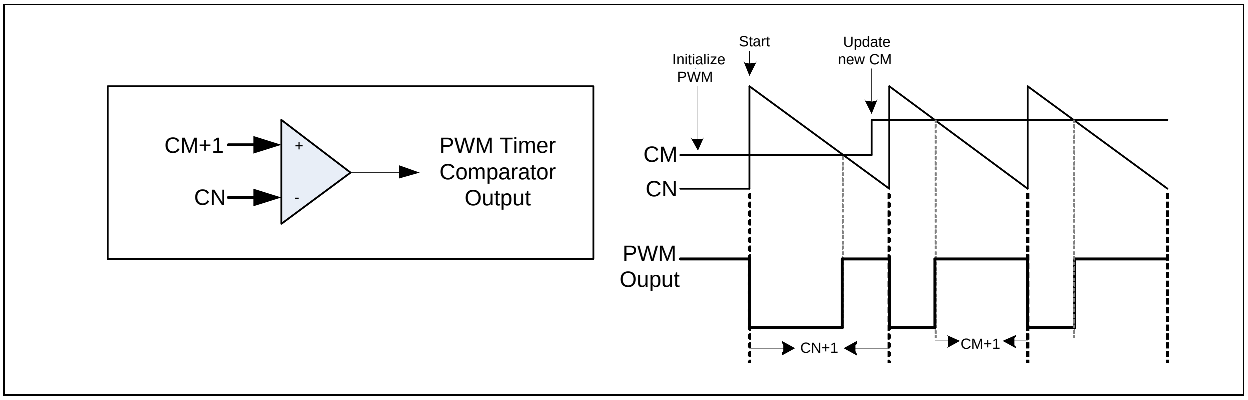


Figure 6-37 Legend of Internal Comparator Output of PWM-Timer

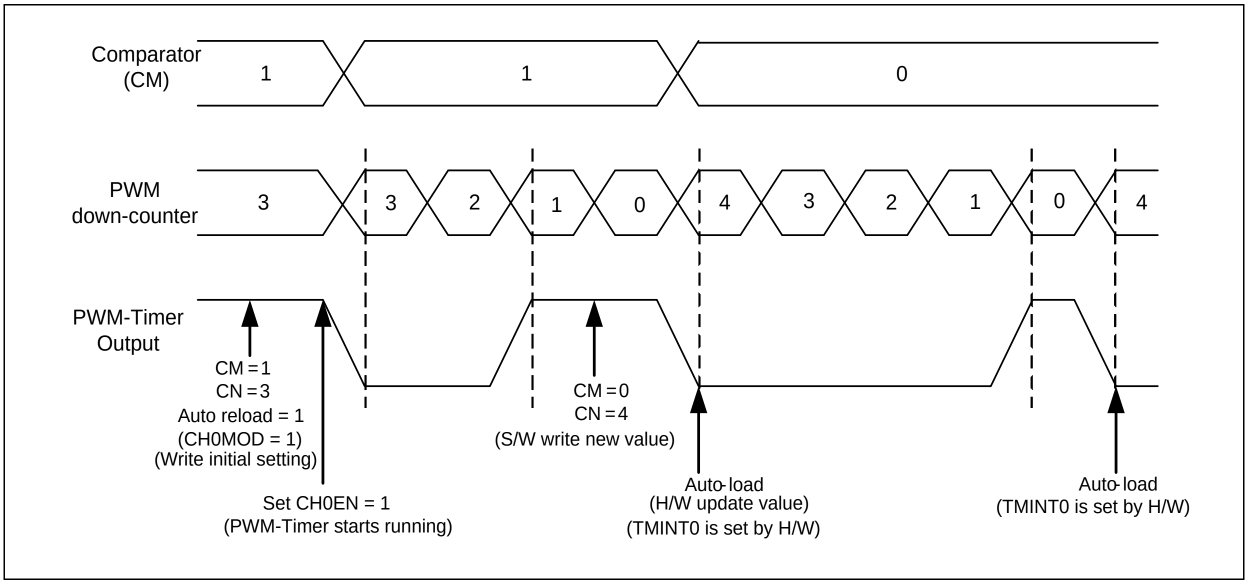


Figure 6-38 PWM-Timer Operation in Edge-aligned Mode

6.10.5.3 Center-aligned Type

For Center-aligned type, the 16 bits PWM-timer is an up-down counter as shown below. The counter of PWM-timer starts counting up from 0 to (CN+1) and then start counting down to 0 to finish a PWM period. The value of PWM counter will be compared with CM (PWM_DUTYy[31:16], y=0~3) to decide the high/low of PWM output.

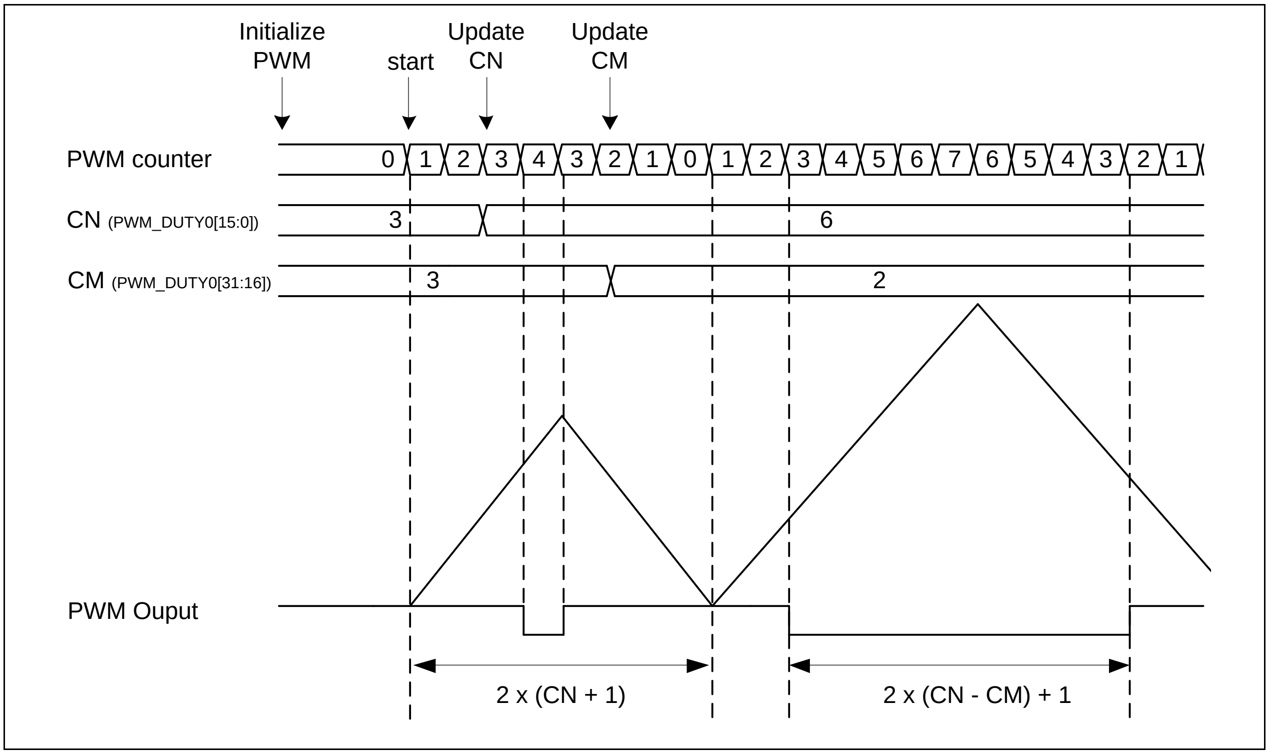


Figure 6‑39 PWM-Timer Operation in Center-aligned Mode

PWM frequency = PWMxy_CLK/(prescale+1)*(clock divider)/(2x(CN+1)); where xy could be 01, 23, depending on selected PWM channel.

Duty ratio = (2xCM+1)/(2x(CN+1)).

CM >= CN: PWM output is always high.

CM < CN: PWM low width = 2x(CN-CM)+1 unit; PWM high width = (2xCM+1) unit.

If CM = 0: PWM low width = (2xCN+1) unit; PWM high width = 1 unit

Note: 1. Unit = one PWM clock cycle.

6.10.5.4 PWM Double Buffering, continuous and one-shot Operation

The PWM has double buffering function; the reload value is updated at the start of next period without affecting current timer operation. The PWM counter value can be written into CN (PWM_DUTYy[15:0], where y=0~3).

The bit CH0MOD(PWM_CTL[3]) in PWM Control Register (PWM_CTL) defines PWM operation in Continuous or One-shot mode If CH0MOD(PWM_CTL[3]) is set to one (continuous mode), the controller loads CN(PWM_DUTY[15:0]) to PWM counter when PWM counter reaches 0. If CN(PWM_DUTY[15:0]) is set to 0, PWM counter will be halt when PWM counter counts to 0.

In one-shot mode (CH0MOD=0; PWM_CTL[3])=0), the corresponding channel will output only one duty waveform and counter will be stopped if no further corresponding duty register updated. When PWM counter is running, updating corresponding duty register will engage the next duty waveform.

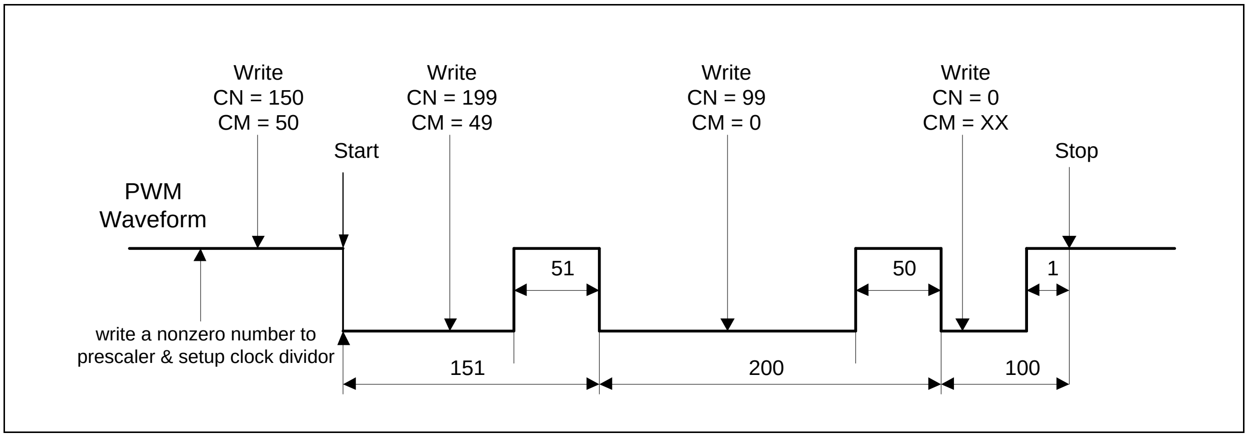


Figure 6-40 PWM Double Buffer Illustration

6.10.5.5 *Modulate Duty Ratio*

The double buffering function allows CM (PWM_DUTYy[31:16]; y=0~3) to be written at any point in current cycle. The loaded value will take effect from next cycle.

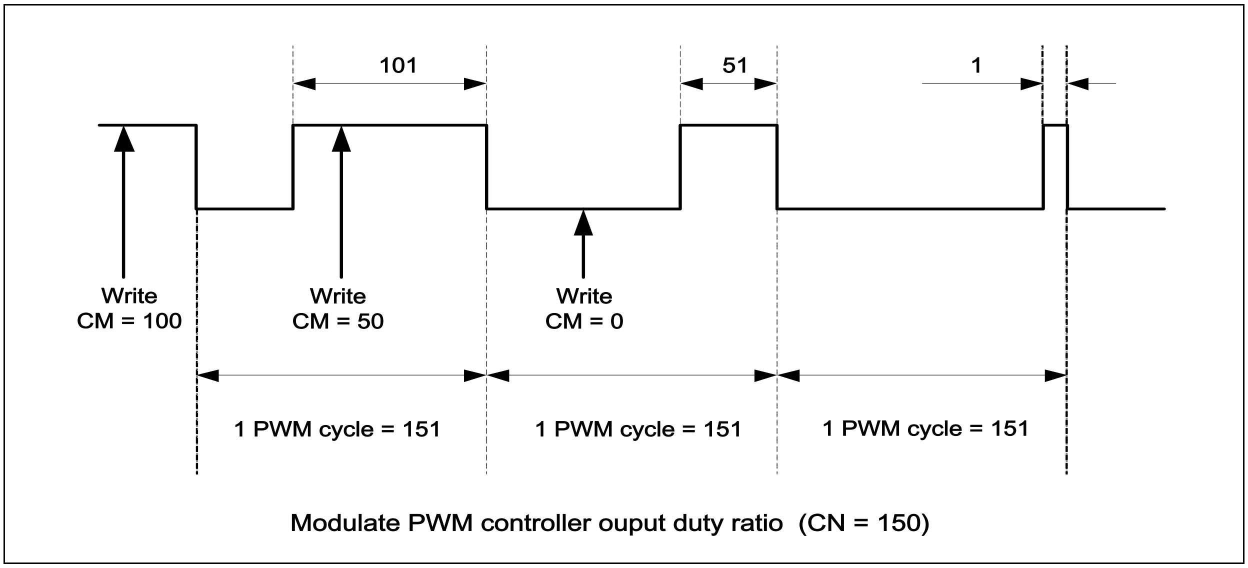


Figure 6-41 PWM Controller Output Duty Ratio

6.10.5.6 *Dead-Zone Generator*

PWM implements Dead Zone generator. They are built for power device protection. This function generates a programmable time gap called “Dead-Zone” to delay PWM rising output, and it is in order to prevent damage for the power switch devices that connected to the PWM output pins. User can program Dead-Zone counter to determine the Dead Zone interval of channel 0, 1 pair with DZ01 (PWM_PRES[23:16]). The Dead Zone period of channel 0, 1 pair can be calculated by (PWM_CLK period x (DZ01 + 1)) and the enable bit is DZEN01 (PWM_CTL[4]).

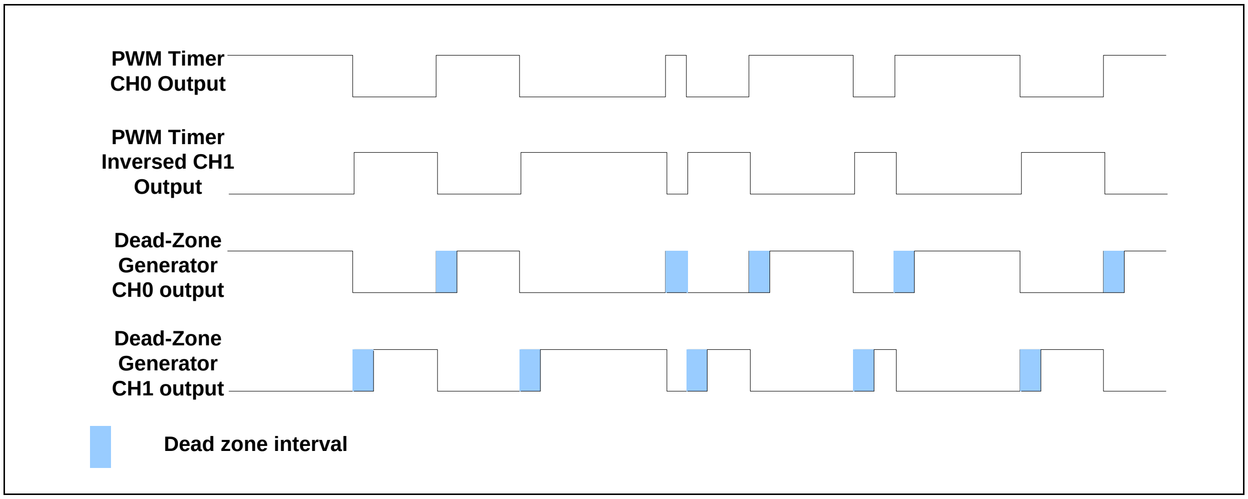

PWM Timer CH0 Output
PWM Timer Inversed CH1 Output
Dead-Zone Generator CH0 output
Dead-Zone Generator CH1 output
Dead zone interval

Figure 6-42 Paired PWM Output with Dead Zone Generation Operation

6.10.5.7 Capture Operation

The Capture channel 0 and PWM channel 0 share one timer ; and the Capture channel1 and PWM channel 1 share another timer, and etc. The capture always latches PWM-timer to CRL (PWM_CRL0[15:0]) when input channel has a rising transition and latches PWM-timer to CFL (PWM_CFL0[15:0]) when input channel has a falling transition. Capture channel 0 interrupt is programmable by setting CRL_IE0 (PWM_CAPINTEN[0]) (Rising latch Interrupt enable) and CFL_IE0 (PWM_CAPINTEN[1]) (Falling latch Interrupt enable) to decide the condition of interrupt occur. Whenever the Capture module issues a capturing flag (rising latched or falling latched) that are defined in PWM_CAPINTSTS, and the reload enable bit (defined in PWM_CAPCTL) is also set the corresponding PWM timer will be reloaded with CN (PWM_DUTYy[15:0], where y=0~3) at this moment. Note that the corresponding I/O pins must be configured as input type before Capture function is enabled.

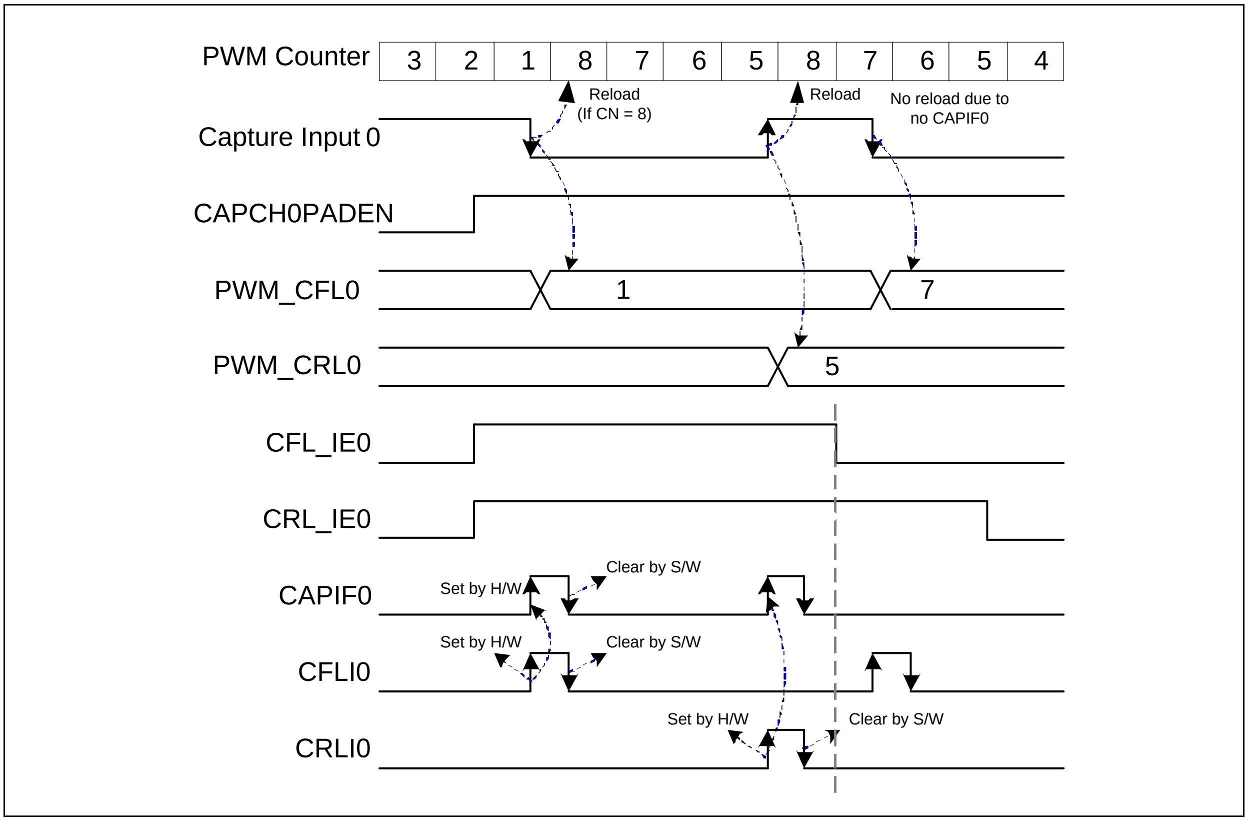


Figure 6-43 PWM Capture Operation Timing

At this example, the CN is 8:

The PWM timer will be reloaded with CN when a capture interrupt flag (CAPIF0) is set.

The channel low pulse width is (CN – CRL), and the channel high pulse width is (CN – CFL).

In some case that need wider counter, user can cascade two 16 bit counters to 32 bit counter for capturing

The cascade method is depicted below

When enabling CH01CASK (PWM_CAPCTL[13]), the internal cascade logic will combine CH0's 16 bit counter with CH1's 16 bit counter to become 32 bit counter for CH0 and the same for enabling CH23CASK (PWM_CAPCTL[29]) for CH2 and CH3. CN0/CN2 (PWM_DUTY0[15:0] / PWM_DUTY2[15:0]) and CN1/CN3 (PWM_DUTY1[15:0] / PWM_DUTY3[15:0]) are also cascaded. At this case, the capturing function for CH1 and CH3 are useless.

When capturing flag is set up, such as rising of CRLI0 (PWM_CAPINTSTS[1]) or falling of CFLI0 (PWM_CAPINTSTS[1]), the capture data for rising latched is stored in CRL0 (PWM_CRL0[15:0]) and CRL1 (PWM_CRL1[15:0]) and CFL0 (PWM_CFL0[15:0]) and CFL1 (PWM_CFL1[15:0]) for falling latched. CRL1 is located in the upper half word and CRL0 is in lower half word and the same for CFL1 and CFL0.

User can also read CRL0 (PWM_CRL[31:0]) or CFL0 (PWM_CFL[31:0]) register directly to get 32-bit capturing data when cascade is enabled

Note: Cascade function is only for PWM capture function.

6.10.5.8 PWM PDMA Function

PWM support PDMA transfer function when operating in capture mode and is only for specified channel (channel 0,2),when the corresponding PDMA enable bit (defined in PWM_CAPCTL register) is set, capture module will issue a request to PDMA controller when the preceding capture event happened. PDMA controller will issue ACK to capture module and read back PDMACH0 register to memory. By setting PDMACAPMOD0 (PWM_CAPCTL[5:4]) and PDMACAPMOD2 (PWM_CAPCTL[21:20]), PDMA can transfer rising latched data or falling latched data or both of them to memory. When using PDMA to transfer both falling and rising data, remember to set CH0RFORDER (PWM_CAPCTL[12]) or CH2RFORDER (PWM_CAPCTL[28]) to decide the order of transferring data (falling edge latched is first or rising edged latched first).

6.10.5.9 PWM-Timer Interrupt Architecture

There are four PWM interrupts, PWMCH0_INT ~ PWMCH3_INT which are OR into PWM_INT. PWM CH0 and Capture channel 0 share one interrupt, PWM CH1 and Capture channel 1 share the same interrupt and so on. Therefore, PWM function and Capture function in the same channel cannot be used at the same time.

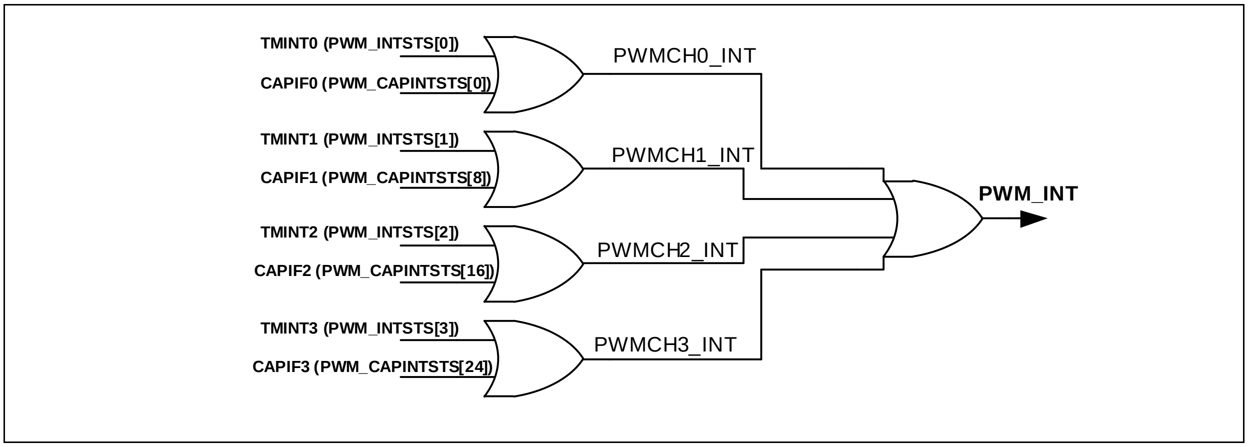


Figure 6-44 PWM-Timer Interrupt

6.10.5.10 PWM-Timer Start Procedure

The following procedure is for starting a PWM drive.

- Set up clock selector (PWM_CLKSEL)
- Set up prescaler (PWM_PRES)
- Set up inverter on/off, dead zone generator on/off, auto-reload/one-shot mode and Stop PWM-timer (PWM_CTL)
- Set up interrupt enable register (PWM_INTEN)
- Set up PWM output enable (PWM_OE)
- Set up the corresponding GPI/O pins to PWM function
- Set up the corresponding GPI/O pins to output type
- Enable PWM down-counter start running (Set ChxEN = 1 in PWM_CTL)
- Set up CM and CN of PWM_DUTYy register for setting PWM duty, y=0~3.(When cascade is enabled the CM is used for the upper half word of the 32 bit CN)

- The procedure 1~8 mentioned above may be set up not in the order and PWM Timer can still work fine

6.10.5.11 PWM-Timer Stop Procedure

Take PWM0, Channel 0 for example:

Method 1:

Set 32 bit PWM_DUTY0 register to 0, and wait for PWM timeout interrupt (if bit 0 of PWM_IE is set) occurring or polling the corresponding time-out flag. When PWM timeout interrupt occurred or the flag is set, disable PWM-Timer (CH0EN; PWM_CTL[0]). (Recommended)

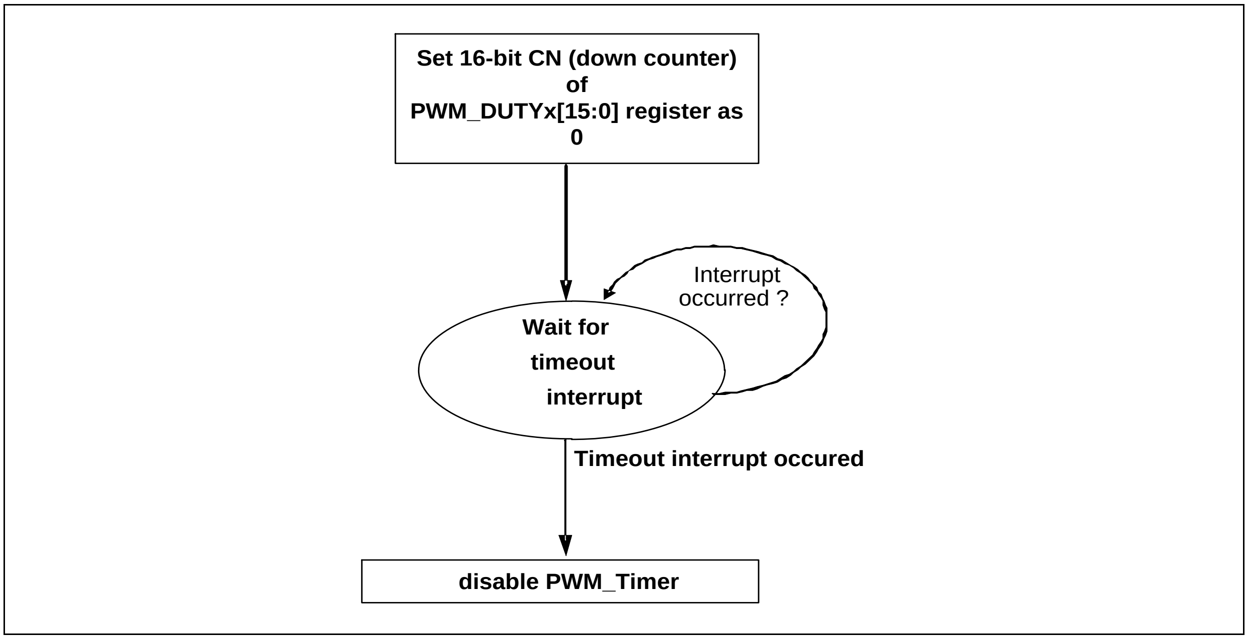

Set 16-bit CN (down counter) of PWM_DUTYx[15:0] register as 0
Interrupt occurred ?
Wait for timeout interrupt
Timeout interrupt occured
disable PWM_Timer

Figure 6-45 PWM-Timer Stop Method 1

Method 2:

Disable PWM-Timer CH0EN (PWM_CTL[0]) directly. (Not recommended)

The reason is that disabling Ch0EN (PWM_CTL[0]) will immediately stop PWM output signal and lead to change the duty of the PWM output, this may cause damage to the control circuit of motor.

6.10.5.12 Capture Start Procedure

- Set up clock selector (PWM_CLKSEL)
- Set up prescaler (PWM_PRES)
- Set up channel enabled, rising/falling interrupt enable and input signal inverter on/off (PWM_CAPCTL, PWM_CAPINTEN)
- Set up PWM down-counter (CN) of PWM_DUTYy register, y=0~3
- Set Capture Input Enable Register (PWM_CAPCTL)
- Set up the corresponding GPI/O pins to PWM function

- Set up the corresponding GPI/O pins to input type
- Enable PWM down-counter start running, i.e. set Ch0EN=1 (PWM_CTL[0]=1)

6.10.6 Register Map

R: read only, **W**: write only, **R/W**: both read and written

Register	Offset	R/W	Description	Reset Value
PWM Base Address: **PWM_BA = 0x4004_0000**				
PWM_PRES	PWM_BA+0x000	R/W	PWM Prescaler Register	0x0000_0000
PWM_CLKSEL	PWM_BA+0x004	R/W	PWM Clock Select Register	0x0000_0000
PWM_CTL	PWM_BA+0x008	R/W	PWM Control Register	0x0000_0000
PWM_INTEN	PWM_BA+0x00C	R/W	PWM Interrupt Enable Register	0x0000_0000
PWM_INTSTS	PWM_BA+0x010	R/W	PWM Interrupt Indication Register	0x0000_0010
PWM_OE	PWM_BA+0x014	R/W	PWM Output Enable for PWM0~PWM3	0x0000_0000
PWM_DUTY0	PWM_BA+0x01C	R/W	PWM Counter/Comparator Register 0	0x0000_0000
PWM_DATA0	PWM_BA+0x020	R	PWM Data Register 0	0x0000_0000
PWM_DUTY1	PWM_BA+0x028	R/W	PWM Counter/Comparator Register 1	0x0000_0000
PWM_DATA1	PWM_BA+0x02C	R	PWM Data Register 1	0x0000_0000
PWM_DUTY2	PWM_BA+0x034	R/W	PWM Counter/Comparator Register 2	0x0000_0000
PWM_DATA2	PWM_BA+0x038	R	PWM Data Register 2	0x0000_0000
PWM_DUTY3	PWM_BA+0x040	R/W	PWM Counter/Comparator Register 3	0x0000_0000
PWM_DATA3	PWM_BA+0x044	R	PWM Data Register 3	0x0000_0000
PWM_CAPCTL	PWM_BA+0x054	R/W	Capture Control Register	0x0000_0000
PWM_CAPINTEN	PWM_BA+0x058	R/W	Capture interrupt enable Register	0x0000_0000
PWM_CAPINTSTS	PWM_BA+0x05C	R/W	Capture Interrupt Indication Register	0x0000_0000
PWM_CRL0	PWM_BA+0x060	R	Capture Rising Latch Register (Channel 0)	0x0000_0000
PWM_CFL0	PWM_BA+0x064	R	Capture Falling Latch Register (Channel 0)	0x0000_0000
PWM_CRL1	PWM_BA+0x068	R	Capture Rising Latch Register (Channel 1)	0x0000_0000
PWM_CFL1	PWM_BA+0x06C	R	Capture Falling Latch Register (Channel 1)	0x0000_0000
PWM_CRL2	PWM_BA+0x070	R	Capture Rising Latch Register (Channel 2)	0x0000_0000
PWM_CFL2	PWM_BA+0x074	R	Capture Falling Latch Register (Channel 2)	0x0000_0000
PWM_CRL3	PWM_BA+0x078	R	Capture Rising Latch Register (Channel 3)	0x0000_0000
PWM_CFL3	PWM_BA+0x07C	R	Capture Falling Latch Register (Channel 3)	0x0000_0000
PWM_PDMACH0	PWM_BA+0x080	R	PDMA Channel 0 Captured Data	0x0000_0000

Register	Offset	R/W	Description	Reset Value
PWM_PDMACH2	PWM_BA+0x084	R	PDMA Channel 2 Captured Data	0x0000_0000
PWM_ADTRGEN	PWM_BA+0x088	R/W	PWM Center-Triggered Control Register	0x0000_0000
PWM_ADTRGSTS	PWM_BA+0x08C	R	PWM Center-Triggered Indication Register	0x0000_0000

6.10.7 Register Description

PWM Pre-scale Register

Register	Offset	R/W	Description	Reset Value
PWM_PRES	PWM_BA+0x000	R/W	PWM Prescaler Register	0x0000_0000

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th></tr>
<tr><td colspan="8">DZ23</td></tr>
<tr><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="8">DZ01</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th></tr>
<tr><td colspan="8">CP23</td></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">CP01</td></tr>
</table>

Bits	Description	
[31:24]	**DZ23**	**Dead Zone Interval Register for CH2 and CH3 Pair** These 8 bits determine dead zone length. The unit time of dead zone length is received from clock selector 2.
[23:16]	**DZ01**	**Dead Zone Interval Register for CH0 and CH1 Pair** These 8 bits determine dead zone length. The unit time of dead zone length is received from clock selector 0.
[15:8]	**CP23**	**Clock Prescaler 2 for PWM Timer 2 & 3** Clock input is divided by (CP23 + 1) before it is fed to the PWM counter 2 & 3 If CP23=0, the prescaler 2 output clock will be stopped. So PWM counter 2 and 3 will be stopped also.
[7:0]	**CP01**	**Clock Prescaler 0 for PWM Timer 0 & 1** Clock input is divided by (CP01 + 1) before it is fed to the PWM counter 0 & 1 If CP01 =0, the prescaler 0 output clock will be stopped. So PWM counter 0 and 1 will be stopped also.

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

PWM Clock Selector Register (PWM_CLKSEL)

Register	Offset	R/W	Description	Reset Value
PWM_CLKSEL	PWM_BA+0x004	R/W	PWM Clock Select Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**CLKSEL3**			**Reserved**	**CLKSEL2**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**CLKSEL1**			**Reserved**	**CLKSEL0**		

Bits	Description	
[31:15]	**Reserved**	Reserved.
[14:12]	**CLKSEL3**	**Timer 3 Clock Source Selection** **Select clock input for timer 3.** 000 = input clock is divided by 2. 001 = input clock is divided by 4. 010 = input clock is divided by 8. 011 = input clock is divided by 16. 100 = input clock is divided by 1.
[10:8]	**CLKSEL2**	**Timer 2 Clock Source Selection** Select clock input for timer 2. (Table is the same as CLKSEL3)
[6:4]	**CLKSEL1**	**Timer 1 Clock Source Selection** Select clock input for timer 1. (Table is the same as CLKSEL3)
[2:0]	**CLKSEL0**	**Timer 0 Clock Source Selection** Select clock input for timer 0. (Table is the same as CLKSEL3)

PWM Control Register (PWM_CTL)

Register	Offset	R/W	Description	Reset Value
PWM_CTL	PWM_BA+0x008	R/W	PWM Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PWMTYPE23	PWMTYPE01	Reserved		CH3MOD	CH3INV	Reserved	CH3EN
23	22	21	20	19	18	17	16
Reserved				CH2MOD	CH2INV	Reserved	CH2EN
15	14	13	12	11	10	9	8
Reserved				CH1MOD	CH1INV	Reserved	CH1EN
7	6	5	4	3	2	1	0
Reserved		DZEN23	DZEN01	CH0MOD	CH0INV	Reserved	CH0EN

Bits	Description	
[31]	**PWMTYPE23**	**Channel 2,3 Counter Mode** 0 = Edge-aligned Mode. 1 = Center-aligned Mode.
[30]	**PWMTYPE01**	**Channel 0,1 Counter Mode** 0 = Edge-aligned Mode. 1 = Center-aligned Mode.
[27]	**CH3MOD**	**PWM-timer 3 Continuous/One-shot Mode** 0 = One-Shot Mode. 1 = Continuous Mode. **Note:** If there is a rising transition at this bit, it will cause CN and CM of PWM0_DUTY3 to be cleared.
[26]	**CH3INV**	**PWM-timer 3 Output Inverter ON/OFF** 0 = Inverter OFF. 1 = Inverter ON.
[24]	**CH3EN**	**PWM-timer 3 Enable/Disable Start Run** 0 = PWM-Timer 3 Running Stopped. 1 = PWM-Timer 3 Start Run Enabled.
[19]	**CH2MOD**	**PWM-timer 2 Continuous/One-shot Mode** 0 = One-Shot Mode. 1 = Continuous Mode. **Note:** If there is a rising transition at this bit, it will cause CN and CM of PWM0_DUTY2 be cleared.
[18]	**CH2INV**	**PWM-timer 2 Output Inverter ON/OFF** 0 = Inverter OFF. 1 = Inverter ON.
[16]	**CH2EN**	**PWM-timer 2 Enable/Disable Start Run**

Bits	Description	
		0 = PWM-Timer 2 Running Stopped. 1 = PWM-Timer 2 Start Run Enabled.
[11]	**CH1MOD**	**PWM-timer 1 Continuous/One-shot Mode** 0 = One-Shot Mode. 1 = Continuous Mode. **Note:** If there is a rising transition at this bit, it will cause CN and CM of PWM0_DUTY1 to be cleared.
[10]	**CH1INV**	**PWM-timer 1 Output Inverter ON/OFF** 0 = Inverter OFF. 1 = Inverter ON.
[8]	**CH1EN**	**PWM-timer 1 Enable/Disable Start Run** 0 = PWM-Timer 1 Running Stopped. 1 = PWM-Timer 1 Start Run Enabled.
[5]	**DZEN23**	**Dead-zone 2 Generator Enable/Disable Control** 0 = Disabled. 1 = Enabled. **Note:** When Dead-Zone Generator is enabled, the pair of PWM2 and PWM3 becomes a complementary pair.
[4]	**DZEN01**	**Dead-zone 0 Generator Enable/Disable Control** 0 = Disabled. 1 = Enabled. **Note:** When Dead-Zone Generator is enabled, the pair of PWM0 and PWM1 becomes a complementary pair.
[3]	**CH0MOD**	**PWM-timer 0 Continuous/One-shot Mode** 0 = One-Shot Mode. 1 = Continuous Mode. **Note:** If there is a rising transition at this bit, it will cause CN and CM of PWM0_DUTY0 to be cleared.
[2]	**CH0INV**	**PWM-timer 0 Output Inverter ON/OFF** 0 = Inverter OFF. 1 = Inverter ON.
[0]	**CH0EN**	**PWM-timer 0 Enable/Disable Start Run** 0 = PWM-Timer 0 Running Stopped. 1 = PWM-Timer 0 Start Run Enabled.

PWM Interrupt Enable Register (PWM_INTEN)

Register	Offset	R/W	Description	Reset Value
PWM_INTEN	PWM_BA+0x00C	R/W	PWM Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				TMIE3	TMIE2	TMIE1	TMIE0

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**TMIE3**	**PWM Timer 3 Interrupt Enable Control** 0 = Disabled. 1 = Enabled.
[2]	**TMIE2**	**PWM Timer 2 Interrupt Enable Control** 0 = Disabled. 1 = Enabled.
[1]	**TMIE1**	**PWM Timer 1 Interrupt Enable Control** 0 = Disabled. 1 = Enabled.
[0]	**TMIE0**	**PWM Timer 0 Interrupt Enable Control** 0 = Disabled. 1 = Enabled.

PWM Interrupt Flag Register (PWM_INTSTS)

Register	Offset	R/W	Description	Reset Value
PWM_INTSTS	PWM_BA+0x010	R/W	PWM Interrupt Indication Register	0x0000_0010

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**PresSyncFlag**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Duty3Syncflag	**Duty2Syncflag**	**Duty1Syncflag**	**Duty0Syncflag**	**TMINT3**	**TMINT2**	**TMINT1**	**TMINT0**

Bits	Description	
[8]	**PresSyncFlag**	**Prescale Synchronize Flag** 0 = Two Prescales have been synchronized to corresponding PWM_CLK (of channel 0,1 or channel 2, 3) domain respectively. 1 = Prescale01 is synchronizing to PWM_CLK domain of channel 0,1 or Prescaler23 is synchronizing to PWM_CLK domain of channel 2, 3. **Note:** software should check this flag when writing Prescale, if this flag is set, and user ignore this flag and change Prescale, the Prescale may be wrong for one prescale cycle
[7]	**Duty3Syncflag**	**Duty3 Synchronize Flag** 0 = Duty3 has been synchronized to PWM_CLK domain of channel 2, 3. 1 = Duty3 is synchronizing to PWM_CLK domain of channel 2, 3. **Note:** software should check this flag when writing duty3, if this flag is set, and user ignore this flag and change duty3, the corresponding CNR and CMR may be wrong for one duty cycle
[6]	**Duty2Syncflag**	**Duty2 Synchronize Flag** 0 = Duty2 has been synchronized to PWM_CLK domain of channel 2, 3. 1 = Duty2 is synchronizing to PWM_CLK domain of channel 2, 3. **Note:** software should check this flag when writing duty2, if this flag is set, and user ignore this flag and change duty2, the corresponding CNR and CMR may be wrong for one duty cycle
[5]	**Duty1Syncflag**	**Duty1 Synchronize Flag** 0 = Duty1 has been synchronized to PWM_CLK domain of channel 0, 1. 1 = Duty1 is synchronizing to PWM_CLK domain of channel 0, 1. **Note:** software should check this flag when writing duty1, if this flag is set, and user ignore this flag and change duty1, the corresponding CNR and CMR may be wrong for one duty cycle
[4]	**Duty0Syncflag**	**Duty0 Synchronize Flag** 0 = Duty0 has been synchronized to PWM_CLK domain of channel 0, 1. 1 = Duty0 is synchronizing to PWM_CLK domain of channel 0, 1. **Note:** software should check this flag when writing duty0, if this flag is set, and user ignore this flag and change duty0, the corresponding CNR and CMR may be wrong for

Bits	Description	
		one duty cycle
[3]	**TMINT3**	**PWM Timer 3 Interrupt Flag** Flag is set by hardware when PWM3 down counter reaches 0, software can clear this bit by writing a one to it.
[2]	**TMINT2**	**PWM Timer 2 Interrupt Flag** Flag is set by hardware when PWM2 down counter reaches 0, software can clear this bit by writing a one to it.
[1]	**TMINT1**	**PWM Timer 1 Interrupt Flag** Flag is set by hardware when PWM1 down counter reaches 0, software can clear this bit by writing a one to it.
[0]	**TMINT0**	**PWM Timer 0 Interrupt Flag** Flag is set by hardware when PWM0 down counter reaches 0, software can clear this bit by writing a one to it.

Note: User can clear each interrupt flag by writing a one to corresponding bit in PWM_INTSTS.

PWM Output Enable Register (PWM_OE)

Register	Offset	R/W	Description	Reset Value
PWM_OE	PWM_BA+0x014	R/W	PWM Output Enable for PWM0~PWM3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**CH3_OE**	**CH2_OE**	**CH1_OE**	**CH0_OE**

Bits	Description	
[3]	**CH3_OE**	**PWM CH3 Output Enable Control** 0 = PWM CH3 output to pin Disabled. 1 = PWM CH3 output to pin Enabled. **Note:** The corresponding GPI/O pin also must be switched to PWM function. (refer to PCx_MFP / PDx_MFP / Pex_MFP / PFx_MFP)
[2]	**CH2_OE**	**PWM CH2 Output Enable Control R** 0 = PWM CH2 output to pin Disabled. 1 = PWM CH2 output to pin Enabled. **Note:** The corresponding GPI/O pin also must be switched to PWM function. (refer to PCx_MFP / PDx_MFP / Pex_MFP / PFx_MFP)
[1]	**CH1_OE**	**PWM CH1 Output Enable Control** 0 = PWM CH1 output to pin Disabled. 1 = PWM CH1 output to pin Enabled. **Note:** The corresponding GPI/O pin also must be switched to PWM function. (refer to PBx_MFP / PCx_MFP / PDx_MFP)
[0]	**CH0_OE**	**PWM CH0 Output Enable Control** 0 = PWM CH0 output to pin Disabled. 1 = PWM CH0 output to pin Enabled. **Note:** The corresponding GPI/O pin also must be switched to PWM function (refer to PBx_MFP / PCx_MFP / PDx_MFP)

PWM DUTY Register 3-0 (PWM_DUTY3~0)

Register	Offset	R/W	Description	Reset Value
PWM_DUTY0	PWM_BA+0x01C	R/W	PWM Counter/Comparator Register 0	0x0000_0000
PWM_DUTY1	PWM_BA+0x028	R/W	PWM Counter/Comparator Register 1	0x0000_0000
PWM_DUTY2	PWM_BA+0x034	R/W	PWM Counter/Comparator Register 2	0x0000_0000
PWM_DUTY3	PWM_BA+0x040	R/W	PWM Counter/Comparator Register 3	0x0000_0000

31	30	29	28	27	26	25	24
CM							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CM							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CN							

Bits	Description	
[31:16]	**CM**	**PWM Comparator Register** CM determines the PWM duty. In edge-aligned mode, PWM frequency = PWMxy_CLK/(prescale+1)*(clock divider)/(CN+1); where xy could be 01, 23, depending on the selected PWM channel. Duty ratio = (CM+1)/(CN+1). CM >= CN: PWM output is always high. CM < CN: PWM low width = (CN-CM) unit; PWM high width = (CM+1) unit. CM = 0: PWM low width = (CN) unit; PWM high width = 1 unit. (Unit = one PWM clock cycle). In center-aligned mode, PWM frequency = PWMxy_CLK/(prescale+1)*(clock divider)/(2x(CN+1)); where xy could be 01, 23, depending on the selected PWM channel. Duty ratio = (2xCM+1)/(2x(CN+1)). CM >= CN: PWM output is always high. CM < CN: PWM low width = 2x(CN-CM)+1 unit; PWM high width = (2xCM+1) unit. CM = 0: PWM low width = (2xCN+1) unit; PWM high width = 1 unit. (Unit = one PWM clock cycle). **Note:**Any write to CM will take effect in next PWM cycle.
[15:0]	**CN**	**PWM Counter/Timer Loaded Value** CN determines the PWM period. In edge-aligned mode, PWM frequency = PWMxy_CLK/(prescale+1)*(clock divider)/(CN+1); where xy could be

Bits	Description	
		01, 23, depending on the selected PWM channel. Duty ratio = (CM+1)/(CN+1). CM >= CN: PWM output is always high. CM < CN: PWM low width = (CN-CM) unit; PWM high width = (CM+1) unit. CM = 0: PWM low width = (CN) unit; PWM high width = 1 unit. (Unit = one PWM clock cycle). In center-aligned mode, PWM frequency = PWMxy_CLK/(prescale+1)*(clock divider)/(2x(CN+1)); where xy could be 01, 23, depending on the selected PWM channel. Duty ratio = (2xCM+1)/(2x(CN+1)). CM >= CN: PWM output is always high. CM < CN: PWM low width = 2x(CN-CM)+1 unit; PWM high width = (2xCM+1) unit. CM = 0: PWM low width = (2xCN+1) unit; PWM high width = 1 unit. (Unit = one PWM clock cycle). **Note:** Any write to CN will take effect in next PWM cycle.

PWM Data Register (PWM_DATA)

Register	Offset	R/W	Description	Reset Value
PWM_DATA0	PWM_BA+0x020	R	PWM Data Register 0	0x0000_0000
PWM_DATA1	PWM_BA+0x02C	R	PWM Data Register 1	0x0000_0000
PWM_DATA2	PWM_BA+0x038	R	PWM Data Register 2	0x0000_0000
PWM_DATA3	PWM_BA+0x044	R	PWM Data Register 3	0x0000_0000

31	30	29	28	27	26	25	24
sync	DATA_H						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DATA_H							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DATA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA							

Bits	Description	
[31]	**sync**	**CN Value Sync with PWM Counter** 0 = CN value is sync to PWM counter. 1 = CN value is not sync to PWM counter. **Note:** when the corresponding cascade enable bit is set, this bit will not appear in the corresponding channel
[30:16]	**DATA_H**	**PWM Data Register** User can monitor PWM_DATA to know the current value in 32-bit down count counter of corresponding channel. **Notes:** This will be valid only for the corresponding cascade enable bit is set
[15:0]	**DATA**	**PWM Data Register** User can monitor PWM_DATA to know the current value in 16-bit down count counter of corresponding channel.

Capture Control Register (PWM_CAPCTL)

Register	Offset	R/W	Description	Reset Value
PWM_CAPCTL	PWM_BA+0x054	R/W	Capture Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CAPRELOADFEN3	CAPRELOADREN3	CH23CASK	CH2RFORDER	Reserved	CAPCH3PADEN	CAPCH3EN	INV3
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CAPRELOADFEN2	CAPRELOADREN2	PDMACAPMOD2		CH2PDMAEN	CAPCH2PADEN	CAPCH2EN	INV2
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CAPRELOADFEN1	CAPRELOADREN1	CH01CASK	CH0RFORDER	Reserved	CAPCH1PADEN	CAPCH1EN	INV1
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CAPRELOADFEN0	CAPRELOADREN0	PDMACAPMOD0		CH0PDMAEN	CAPCH0PADEN	CAPCH0EN	INV0

Bits	Description	
[31]	**CAPRELOADFEN3**	**Reload CNR3 When CH3 Falling Capture Event Comes** 0 = Falling capture reload for CH3 Disabled. 1 = Falling capture reload for CH3 Enabled.
[30]	**CAPRELOADREN3**	**Reload CNR3 When CH3 Rising Capture Event Comes** 0 = Rising capture reload for CH3 Disabled. 1 = Rising capture reload for CH3 Enabled.
[29]	**CH23CASK**	**Cascade channel 2 and channel 3 PWM counter for capturing usage**
[28]	**CH2RFORDER**	**Set this bit to determine whether the PWM_CRL2 or PWM_CFL2 is the first captured data transferred to memory through PDMA when PDMACAPMOD2 = 2'b11.** 0 = PWM_CFL2 is the first captured data to memory. 1 = PWM_CRL2 is the first captured data to memory.
[27]	**Reserved**	Reserved.
[26]	**CAPCH3PADEN**	**Capture Input Enable Control** 0 = Disable the channel 3 input capture signal from corresponding multi-function pin. 1 = Enable the channel 3 input capture signal from corresponding multi-function pin.
[25]	**CAPCH3EN**	**Capture Channel 3 Transition Enable/Disable Control** 0 = Capture function on channel 3 Disabled. 1 = Capture function on channel 3 Enabled. When Enabled, Capture latched the PMW-timer and saved to CRL3 (PWM_CRL3[15:0]) for rising latch and CFL3 (PWM_CFL3[15:0]) for falling latch. When Disabled, Capture does not update CRL3 (PWM_CRL3[15:0]) and CFL3 (PWM_CFL3[15:0]), and disable Channel 3 Interrupt.
[24]	**INV3**	**Channel 3 Inverter ON/OFF** 0 = Inverter OFF.

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

Bits	Description	
		1 = Inverter ON. Reverse the input signal from GPIO before fed to Capture timer
[23]	**CAPRELOADFEN2**	**Reload CNR2 When CH2 Capture Failing Event Coming** 0 = Failing capture reload for CH2 Disabled. 1 = Failing capture reload for CH2 Enabled.
[22]	**CAPRELOADREN2**	**Reload CNR2 When CH2 Capture Rising Event Coming** 0 = Rising capture reload for CH2 Disabled. 1 = Rising capture reload for CH2 Enabled.
[21:20]	**PDMACAPMOD2**	**Select CRL2 or CFL2 for PDMA Transfer** 00 = reserved. 01 = CRL2 will be transmitted. 10 = CFL2 will be transmitted. 11 = Both CRL2 and CFL2 will be transmitted.
[19]	**CH2PDMAEN**	**Channel 2 PDMA Enable Control** 0 = Channel 2 PDMA function Disabled. 1 = Channel 2 PDMA function Enabled for the channel 2 captured data and transfer to memory.
[18]	**CAPCH2PADEN**	**Capture Input Enable Control** 0 = Disable the channel 2 input capture signal from corresponding multi-function pin. 1 = Enable the channel 2 input capture signal from corresponding multi-function pin.
[17]	**CAPCH2EN**	**Capture Channel 2 Transition Enable/Disable Control** 0 = Capture function on channel 2 Disabled. 1 = Capture function on channel 2 Enabled. When Enabled, Capture latched the PWM-timer value and saved to CRL2 (PWM_CRL2[15:0]) for rising latch and CFL2 (PWM_CFL2[15:0]) for falling latch. When Disabled, Capture does not update CRL2 (PWM_CRL2[15:0]) and CFL2 (PWM_CFL2[15:0]), and disable Channel 2 Interrupt.
[16]	**INV2**	**Channel 2 Inverter ON/OFF** 0 = Inverter OFF. 1 = Inverter ON. Reverse the input signal from GPIO before fed to Capture timer
[15]	**CAPRELOADFEN1**	**Reload CNR1 When CH1 Capture Falling Event Coming** 0 = Capture falling reload for CH1 Disabled. 1 = Capture falling reload for CH1 Enabled.
[14]	**CAPRELOADREN1**	**Reload CNR1 When CH1 Capture Rising Event Comes** 0 = Rising capture reload for CH1 Disabled. 1 = Rising capture reload for CH1 Enabled.
[13]	**CH01CASK**	**Cascade channel 0 and channel 1 PWM timer for capturing usage**
[12]	**CH0RFORDER**	**Set this bit to determine whether the PWM_CRL0 or PWM_CFL0 is the first captured data transferred to memory through PDMA when PDMACAPMOD0 =2'b11.** 0 = PWM_CFL0 is the first captured data to memory. 1 = PWM_CRL0 is the first captured data to memory.
[11]	**Reserved**	Reserved.
[10]	**CAPCH1PADEN**	**Capture Input Enable Control** 0 = Disable the channel 1 input capture signal from corresponding multi-function pin. 1 = Enable the channel 1 input capture signal from corresponding multi-function pin.

Bits	Description	
[9]	**CAPCH1EN**	**Capture Channel 1 Transition Enable/Disable Control** 0 = Capture function on channel 1 Disabled. 1 = Capture function on channel 1 Enabled. When Enabled, Capture latched the PMW-counter and saved to CRL1 (PWM_CRL1[15:0]) for rising latch and CFL1 (PWM_CFL1[15:0]) for falling latch. When Disabled, Capture does not update CRL1 (PWM_CRL1[15:0]) and CFL1 (PWM_CFL1[15:0]), and disable Channel 1 Interrupt.
[8]	**INV1**	**Channel 1 Inverter ON/OFF** 0 = Inverter OFF. 1 = Inverter ON. Reverse the input signal from GPIO before fed to Capture timer
[7]	**CAPRELOADFEN0**	**Reload CNR0 When CH0 Capture Falling Event Comes** 0 = Falling capture reload for CH0 Disabled. 1 = Falling capture reload for CH0 Enabled.
[6]	**CAPRELOADREN0**	**Reload CNR0 When CH0 Capture Rising Event Comes** 0 = Rising capture reload for CH0 Disabled. 1 = Rising capture reload for CH0 Enabled.
[5:4]	**PDMACAPMOD0**	**Select CRL0 or CFL0 for PDMA Transfer** 00 = reserved. 01 = CRL0 will be transmitted. 10 = CFL0 will be transmitted. 11 = Both CRL0 and CFL0 will be transmitted.
[3]	**CH0PDMAEN**	**Channel 0 PDMA Enable Control** 0 = Channel 0 PDMA function Disabled. 1 = Channel 0 PDMA function Enabled for the channel 0 captured data and transfer to memory.
[2]	**CAPCH0PADEN**	**Capture Input Enable Control** 0 **=** Disable the channel 0 input capture signal from corresponding multi-function pin. 1 = Enable the channel 0 input capture signal from corresponding multi-function pin.
[1]	**CAPCH0EN**	**Capture Channel 0 Transition Enable/Disable Control** 0 = Capture function on channel 0 Disabled. 1 = Capture function on channel 0 Enabled. When Enabled, Capture latched the PWM-timer value and saved to CRL0 (PWM_CRL0[15:0]) for rising latch and CFL0 (PWM_CFL0[15:0]) for falling latch. When Disabled, Capture does not update CRL0 (PWM_CRL0[15:0]) and CFL0 (PWM_CFL0[15:0]), and disable Channel 0 Interrupt.
[0]	**INV0**	**Channel 0 Inverter ON/OFF** 0 = Inverter OFF. 1 = Inverter ON. Reverse the input signal from GPIO before fed to Capture timer

Capture Interrupt enable Register (PWM_CAPINTEN)

Register	Offset	R/W	Description	Reset Value
PWM_CAPINTEN	PWM_BA+0x058	R/W	Capture interrupt enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						CFL_IE3	CRL_IE3
23	22	21	20	19	18	17	16
Reserved						CFL_IE2	CRL_IE2
15	14	13	12	11	10	9	8
Reserved						CFL_IE1	CRL_IE1
7	6	5	4	3	2	1	0
Reserved						CFL_IE0	CRL_IE0

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25]	**CFL_IE3**	**Channel 3 Falling Latch Interrupt Enable ON/OFF** 0 = Falling latch interrupt Disabled. 1 = Falling latch interrupt Enabled. When Enabled, if Capture detects Channel 3 has falling transition, Capture issues an Interrupt.
[24]	**CRL_IE3**	**Channel 3 Rising Latch Interrupt Enable ON/OFF** 0 = Rising latch interrupt Disabled. 1 = Rising latch interrupt Enabled. When Enabled, if Capture detects Channel 3 has rising transition, Capture issues an Interrupt.
[23:18]	**Reserved**	Reserved.
[17]	**CFL_IE2**	**Channel 2 Falling Latch Interrupt Enable ON/OFF** 0 = Falling latch interrupt Disabled. 1 = Falling latch interrupt Enabled. When Enabled, if Capture detects Channel 2 has falling transition, Capture issues an Interrupt.
[16]	**CRL_IE2**	**Channel 2 Rising Latch Interrupt Enable ON/OFF** 0 = Rising latch interrupt Disabled. 1 = Rising latch interrupt Enabled. When Enabled, if Capture detects Channel 2 has rising transition, Capture issues an Interrupt.
[15:10]	**Reserved**	Reserved.
[9]	**CFL_IE1**	**Channel 1 Falling Latch Interrupt Enable Control** 0 = Falling latch interrupt Disabled. 1 = Falling latch interrupt Enabled. When Enabled, if Capture detects Channel 1 has falling transition, Capture issues an

Bits	Description	
		Interrupt.
[8]	**CRL_IE1**	**Channel 1 Rising Latch Interrupt Enable Control** 0 = Rising latch interrupt Disabled. 1 = Rising latch interrupt Enabled. When Enabled, if Capture detects Channel 1 has rising transition, Capture issues an Interrupt.
[7:2]	**Reserved**	Reserved.
[1]	**CFL_IE0**	**Channel 0 Falling Latch Interrupt Enable ON/OFF** 0 = Falling latch interrupt Disabled. 1 = Falling latch interrupt Enabled. When Enabled, if Capture detects Channel 0 has falling transition, Capture issues an Interrupt.
[0]	**CRL_IE0**	**Channel 0 Rising Latch Interrupt Enable ON/OFF** 0 = Rising latch interrupt Disabled. 1 = Rising latch interrupt Enabled. When Enabled, if Capture detects Channel 0 has rising transition, Capture issues an Interrupt.

Capture Interrupt Status Register (PWM_CAPINTSTS)

Register	Offset	R/W	Description	Reset Value
PWM_CAPINTSTS	PWM_BA+0x05C	R/W	Capture Interrupt Indication Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			CAPOVF3	CAPOVR3	CFLI3	CRLI3	CAPIF3
23	22	21	20	19	18	17	16
Reserved			CAPOVF2	CAPOVR2	CFLI2	CRLI2	CAPIF2
15	14	13	12	11	10	9	8
Reserved			CAPOVF1	CAPOVR1	CFLI1	CRLI1	CAPIF1
7	6	5	4	3	2	1	0
Reserved			CAPOVF0	CAPOVR0	CFLI0	CRLI0	CAPIF0

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28]	**CAPOVF3**	**Capture Falling Flag Over Run for Channel 3** This flag indicate CFL3 update faster than software reading it when it is set This bit will be cleared automatically when user clear CFLI3 (PWM_CAPINTSTS[26])
[27]	**CAPOVR3**	**Capture Rising Flag Over Run for Channel 3** This flag indicate CRL3update faster than software reading it when it is set This bit will be cleared automatically when user clear CRLI3 (PWM_CAPINTSTS[25])
[26]	**CFLI3**	**PWM_CFL3 Latched Indicator Bit** When input channel 3 has a falling transition, PWM_CFL3 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing 1 to it.
[25]	**CRLI3**	**PWM_CRL3 Latched Indicator Bit** When input channel 3 has a rising transition, PWM_CRL3 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing 1 to it.
[24]	**CAPIF3**	**Capture3 Interrupt Indication Flag** If channel 3 rising latch interrupt (CRL_IE3, PWM_CAPINTEN[24]) is enabled, a rising transition occurs at input channel 3 will result in CAPIF3 to high; Similarly, a falling transition will cause CAPIF3 to be set high if channel 3 falling latch interrupt (CFL_IE3, PWM_CAPINTEN[25]) is enabled. This flag is cleared by software with a write 1 on it.
[23:21]	**Reserved**	Reserved.
[20]	**CAPOVF2**	**Capture Falling Flag Over Run for Channel 2** This flag indicate CFL2 update faster than software reading it when it is set This bit will be cleared automatically when user clear CFLI2 (PWM_CAPINTSTS[18])
[19]	**CAPOVR2**	**Capture Rising Flag Over Run for Channel 2** This flag indicate CRL2 update faster than software reading it when it is set This bit will be cleared automatically when user clear CRLI2 (PWM_CAPINTSTS[17])

Bits	Description	
[18]	**CFLI2**	**PWM_CFL2 Latched Indicator Bit** When input channel 2 has a falling transition, PWM0_CFL2 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing 1 to it.
[17]	**CRLI2**	**PWM_CRL2 Latched Indicator Bit** When input channel 2 has a rising transition, PWM0_CRL2 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing 1 to it.
[16]	**CAPIF2**	**Capture2 Interrupt Indication Flag** If channel 2 rising latch interrupt (CRL_IE2, PWM_CAPINTEN[16]) is enabled, a rising transition occurs at input channel 2 will result in CAPIF2 to high; Similarly, a falling transition will cause CAPIF2 to be set high if channel 2 falling latch interrupt (CFL_IE2, PWM_CAPINTEN[17]) is enabled. This flag is cleared by software with a write 1 on it.
[15:13]	**Reserved**	Reserved.
[12]	**CAPOVF1**	**Capture Falling Flag Over Run for Channel 1** This flag indicate CFL1 update faster than software reading it when it is set This bit will be cleared automatically when user clear CFLI1 (PWM_CAPINTSTS[10])
[11]	**CAPOVR1**	**Capture Rising Flag Over Run for Channel 1** This flag indicate CRL1 update faster than software reading it when it is set This bit will be cleared automatically when user clear CRLI1 (PWM_CAPINTSTS[9])
[10]	**CFLI1**	**PWM_CFL1 Latched Indicator Bit** When input channel 1 has a falling transition, PWM_CFL1 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing 1 to it.
[9]	**CRLI1**	**PWM_CRL1 Latched Indicator Bit** When input channel 1 has a rising transition, PWM_CRL1 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing 1 to it.
[8]	**CAPIF1**	**Capture1 Interrupt Indication Flag** If channel 1 rising latch interrupt (CRL_IE1, PWM_CAPINTEN[8]) is enabled, a rising transition occurs at input channel 1 will result in CAPIF1 to high; Similarly, a falling transition will cause CAPIF1 to be set high if channel 1 falling latch interrupt (CFL_IE1, PWM_CAPINTEN[9]) is enabled. This flag is cleared by software with a write 1 on it.
[7:5]	**Reserved**	Reserved.
[4]	**CAPOVF0**	**Capture Falling Flag Over Run for Channel 0** This flag indicate CFL0 update faster than software read it when it is set This bit will be cleared automatically when user clear CFLI0 (PWM_CAPINTSTS[2])
[3]	**CAPOVR0**	**Capture Rising Flag Over Run for Channel 0** This flag indicate CRL0 update faster than software reading it when it is set This bit will be cleared automatically when user clears CRLI0 (PWM_CAPINTSTS[1]).
[2]	**CFLI0**	**PWM_CFL0 Latched Indicator Bit** When input channel 0 has a falling transition, PWM0_CFL0 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing 1 to it.
[1]	**CRLI0**	**PWM_CRL0 Latched Indicator Bit** When input channel 0 has a rising transition, PWM0_CRL0 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing 1 to it.

Bits	Description	
[0]	**CAPIF0**	**Capture0 Interrupt Indication Flag** If channel 0 rising latch interrupt (CRL_IE0, PWM_CAPINTEN[0]) is enabled, a rising transition occurs at input channel 0 will result in CAPIF0 to high; Similarly, a falling transition will cause CAPIF0 to be set high if channel 0 falling latch interrupt (CFL_IE0, PWM_CAPINTEN[1]) is enabled. This flag is cleared by software with a write 1 on it.

Capture Rising Latch Register3-0 (PWM_CRL3-0)

Register	Offset	R/W	Description	Reset Value
PWM_CRL0	PWM_BA+0x060	R	Capture Rising Latch Register (Channel 0)	0x0000_0000
PWM_CRL1	PWM_BA+0x068	R	Capture Rising Latch Register (Channel 1)	0x0000_0000
PWM_CRL2	PWM_BA+0x070	R	Capture Rising Latch Register (Channel 2)	0x0000_0000
PWM_CRL3	PWM_BA+0x078	R	Capture Rising Latch Register (Channel 3)	0x0000_0000

31	30	29	28	27	26	25	24
CRL_H							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CRL_H							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CRL							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRL							

Bits	Description	
[31:16]	**CRL_H**	**Upper Half Word of 32-bit Capture Data When Cascade Enable Control** When cascade is enabled for capture channel 0, 2,the original 16 bit counter extend to 32 bit, and capture result CRL0 and CRL2 are also extend to 32 bit,
[15:0]	**CRL**	**Capture Rising Latch Register** Latch the PWM counter when Channel 0/1/2/3 has rising transition.

PWM Capture Falling Latch Register3-0 (PWM_CFL3-0)

Register	Offset	R/W	Description	Reset Value
PWM_CFL0	PWM_BA+0x064	R	Capture Falling Latch Register (Channel 0)	0x0000_0000
PWM_CFL1	PWM_BA+0x06C	R	Capture Falling Latch Register (Channel 1)	0x0000_0000
PWM_CFL2	PWM_BA+0x074	R	Capture Falling Latch Register (Channel 2)	0x0000_0000
PWM_CFL3	PWM_BA+0x07C	R	Capture Falling Latch Register (Channel 3)	0x0000_0000

31	30	29	28	27	26	25	24
CFL_H							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CFL_H							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CFL							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CFL							

Bits	Description	
[31:16]	**CFL_H**	**Upper Half Word of 32-bit Capture Data When Cascade Enable Control** When cascade is enabled for capture channel 0, 2, the original 16 bit counter will be extended to 32 bit, and capture result CFL0 and CFL2 are also extended to 32 bit,
[15:0]	**CFL**	**Capture Falling Latch Register** Latch the PWM counter when Channel 0/1/2/3 has Falling transition.

PDMA Data Register for Capture Channel 0 (PWM_PDMACH0)

Register	Offset	R/W	Description	Reset Value
PWM_PDMACH0	PWM_BA+0x080	R	PDMA Channel 0 Captured Data	0x0000_0000

31	30	29	28	27	26	25	24
PDMACH04							
23	22	21	20	19	18	17	16
PDMACH03							
15	14	13	12	11	10	9	8
PDMACH02							
7	6	5	4	3	2	1	0
PDMACH01							

Bits	Description	
[31:24]	**PDMACH04**	**Captured Data of Channel 0** When CH01CASK is disabled, this byte is 0 When CH01CASK is enabled, It is the 4th byte of 32 bit capturing data for channel 0. **Note:** This register is used as a buffer to transfer PWM captured rising or falling data to memory by PDMA.
[23:16]	**PDMACH03**	**Captured Data of Channel 0** When CH01CASK is disabled, this byte is 0 When CH01CASK is enabled, It is the 3rd byte of 32 bit capturing data for channel 0. **Note:** This register is used as a buffer to transfer PWM captured rising or falling data to memory by PDMA.
[15:8]	**PDMACH02**	**Captured Data of Channel 0** When CH01CASK is disabled, this byte is 0 When CH01CASK is enabled, It is the 2nd byte of 32 bit capturing data for channel 0. **Note:** This register is used as a buffer to transfer PWM captured rising or falling data to memory by PDMA.
[7:0]	**PDMACH01**	**Captured Data of Channel 0** When CH01CASK is disabled, this byte is 0 When CH01CASK is enabled, It is the 1st byte of 32 bit capturing data for channel 0. **Note:** This register is used as a buffer to transfer PWM captured rising or falling data to memory by PDMA.

PDMA Data Register for Capture Channel 2 (PWM_PDMACH2)

Register	Offset	R/W	Description	Reset Value
PWM_PDMACH2	PWM_BA+0x084	R	PDMA Channel 2 Captured Data	0x0000_0000

31	30	29	28	27	26	25	24
PDMACH24							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PDMACH23							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PDMACH22							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDMACH21							

Bits	Description	
[31:24]	**PDMACH24**	**Captured Data of Channel 2** When CH23CASK is disabled, this byte is 0 When CH23CASK is enabled, It is the 4th byte of 32 bit capturing data for channel 0. **Note:** This register is used as a buffer to transfer PWM captured rising or falling data to memory by PDMA.
[23:16]	**PDMACH23**	**Captured Data of Channel 2** When CH23CASK is disabled, this byte is 0 When CH23CASK is enabled, It is the 3rd byte of 32 bit capturing data for channel 0. **Note:** This register is used as a buffer to transfer PWM captured rising or falling data to memory by PDMA.
[15:8]	**PDMACH22**	**Captured Data of Channel 2** When CH23CASK is disabled, this byte is 0 When CH23CASK is enabled, It is the 2nd byte of 32 bit capturing data for channel 0. **Note:** This register is used as a buffer to transfer PWM captured rising or falling data to memory by PDMA.
[7:0]	**PDMACH21**	**Captured Data of Channel 2** When CH23CASK is disabled, this byte is 0 When CH23CASK is enabled, It is the 1st byte of 32 bit capturing data for channel 0. **Note:** This register is used as a buffer to transfer PWM captured rising or falling data to memory by PDMA.

PWM Center-Triggered Control Register (PWM_ADTRGEN)

Register	Offset	R/W	Description	Reset Value
PWM_ADTRGEN	PWM_BA+0x088	R/W	PWM Center-Triggered Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**TRGCH3EN**	**TRGCH2EN**	**TRGCH1EN**	**TRGCH0EN**

Bits	Description	
[3]	**TRGCH3EN**	**PWM CH3 Center-triggered Enable Control** 0 = PWM CH3 center-triggered function Disabled. 1 = PWM CH3 center-triggered function Enabled. **Note:** The center-triggered function is only valid in PWM center-aligned mode.
[2]	**TRGCH2EN**	**PWM CH2 Center-triggered Enable Control** 0 = PWM CH2 center-triggered function Disabled. 1 = PWM CH2 center-triggered function Enabled. **Note:** The center-triggered function is only valid in PWM center-aligned mode.
[1]	**TRGCH1EN**	**PWM CH1 Center-triggered Enable Control** 0 = PWM CH1 center-triggered function Disabled. 1 = PWM CH1 center-triggered function Enabled. **Note:** The center-triggered function is only valid in PWM center-aligned mode.
[0]	**TRGCH0EN**	**PWM CH0 Center-triggered Enable Control** 0 = PWM CH0 center-triggered function Disabled. 1 = PWM CH0 center-triggered function Enabled. **Note:** The center-triggered function is only valid in PWM center-aligned mode.

PWM Center-Triggered Indication Register (PWM_ADTRGSTS)

Register	Offset	R/W	Description	Reset Value
PWM_ADTRGSTS	PWM_BA+0x08C	R	PWM Center-Triggered Indication Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**ADTRG3Flag**	**ADTRG2Flag**	**ADTRG1Flag**	**ADTRG0Flag**

Bits	Description	
[3]	**ADTRG3Flag**	**PWM CH3 Center-triggered Flag** 0 = PWM CH3 has not crossed half of PWM period yet. 1 = PWM CH3 has crossed half of PWM period. **Note:** This flag is only valid in center-aligned mode, and software could write 1 into this bit to clear the flag
[2]	**ADTRG2Flag**	**PWM CH2 Center-triggered Flag** 0 = PWM CH2 has not crossed half of PWM period yet. 1 = PWM CH2 has crossed half of PWM period. **Note:** This flag is only valid in center-aligned mode, and software could write 1 into this bit to clear the flag
[1]	**ADTRG1Flag**	**PWM CH1 Center-triggered Flag** 0 = PWM CH1 has not crossed half of PWM period yet. 1 = PWM CH1 has crossed half of PWM period. **Note:** This flag is only valid in center-aligned mode, and software could write 1 into this bit to clear the flag
[0]	**ADTRG0Flag**	**PWM CH0 Center-triggered Flag** 0 = PWM CH0 has not crossed half of PWM period yet. 1 = PWM CH0 has crossed half of PWM period. **Note:** This flag is only valid in center-aligned mode, and software could write 1 into this bit to clear the flag

6.11 Watchdog Timer Controller

6.11.1 Overview

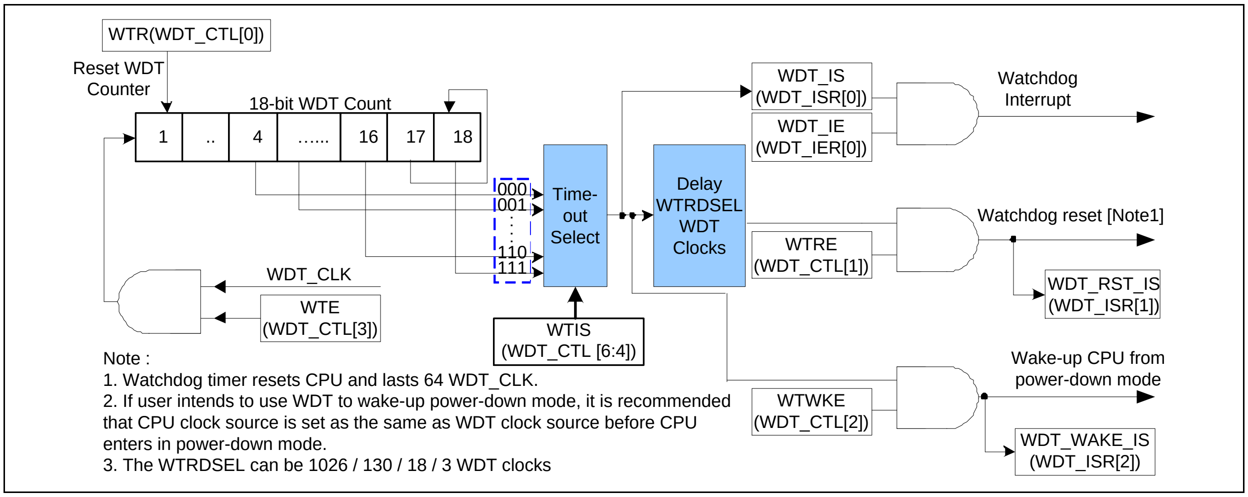
The purpose of Watchdog Timer is to perform a system reset after the software running into a problem. This prevents system from hanging for an infinite period of time. Besides, this Watchdog Timer supports the function to wake-up CPU from Power-down mode. The watchdog timer includes an 18-bit free running counter with programmable time-out intervals.

6.11.2 Features

- 18-bit free running WDT counter for Watchdog timer time-out interval.
- Selectable time-out interval (2^4 ~ 2^{18}) and the time-out interval is 104 ms ~ 26.316 s (if WDT_CLK = 10 kHz).
- Reset period = (1 / 10 kHz) * 63, if WDT_CLK = 10 kHz.

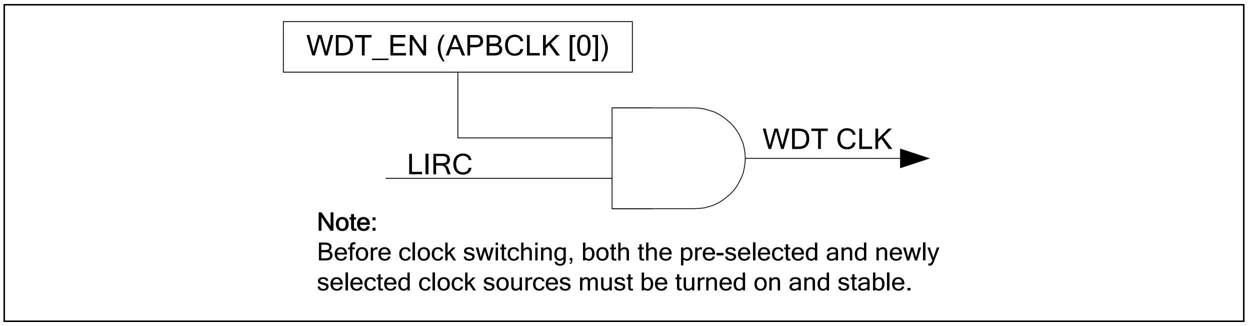
6.11.3 Block Diagram

The Watchdog Timer clock control and block diagram are shown as follows.



Note :
1. Watchdog timer resets CPU and lasts 64 WDT_CLK.
2. If user intends to use WDT to wake-up power-down mode, it is recommended that CPU clock source is set as the same as WDT clock source before CPU enters in power-down mode.
3. The WTRDSEL can be 1026 / 130 / 18 / 3 WDT clocks

Figure 6-46 Watchdog Controller Block Diagram



Note:
Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.

Figure 6-47 Watchdog Timer Clock Control Diagram

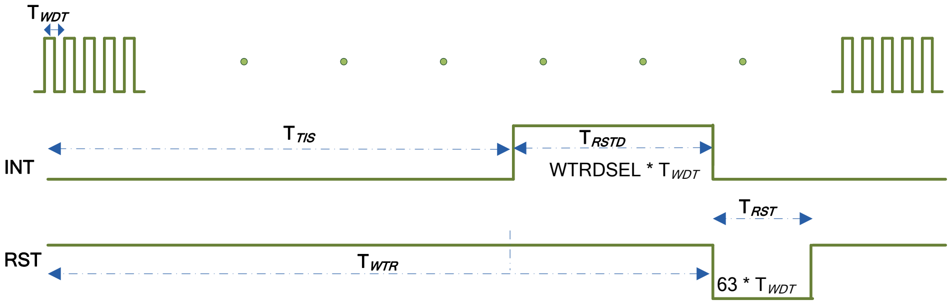
6.11.4 Functional Description

The purpose of Watchdog Timer is to perform a system reset after the software running into a problem. This prevents system from hanging for an infinite period of time. Besides, this Watchdog Timer supports the function to wake-up the chip from Power-down mode. Moreover, the Watchdog counter will be automatically reset when the chip is entering Power-down mode. The watchdog timer includes an 18-bit free running counter with programmable time-out intervals. The following figure show the watchdog time-out interval selection and the next figure show the timing of watchdog interrupt signal and reset signal.

Setting WTE (WDT_CTL [3]) enables the watchdog timer and the WDT counter starts counting up. When the counter reaches the selected time-out interval, Watchdog timer interrupt flag WDT_IS (WDT_ISR[0]) will be set immediately to request a WDT interrupt if the watchdog timer interrupt enable bit WDT_IE (WDT_IER[0]) is set, in the meanwhile, a specified delay time (WTRDSEL (WDT_CTL[9:8]) * T^{WDT}) follows the time-out event. User must set WTR (WDT_CTL [0]) (Watchdog timer reset) high to reset the 18-bit WDT counter to avoid CPU from Watchdog timer reset before the delay time expires. WTR (WDT_CTL [0]) bit is auto cleared by hardware after WDT counter is reset. There are eight time-out intervals with specific delay time which are selected by Watchdog timer interval select bits WTIS (WDT_CTL [6:4]). If the WDT counter has not been cleared after the specific delay time expires, the watchdog timer will set Watchdog Timer Reset Flag, WDT_RST_IS (WDT_ISR[1]) high and reset CPU. This reset will last 63 * WDT clocks (T^{RST}) then CPU restarts executing program from reset vector (0x0000 0000). WDT_RST_IS will not be cleared by Watchdog reset. User may poll WDT_RST_IS by software to recognize the reset source.

WTIS	WTR Time-Out Interval	Reset Delay Period T_{RSTD}	Time-Out Interval WDT_CLK = 10 KHz T_{TIS}
000	2^4 * T_{WDT}	(3/18/130/1026) * T_{WDT}	1.6 ms
001	2^6 * T_{WDT}	(3/18/130/1026) * T_{WDT}	6.4 ms
010	2^8 * T_{WDT}	(3/18/130/1026) * T_{WDT}	25.6 ms
011	2^{10} * T_{WDT}	(3/18/130/1026) * T_{WDT}	102.4 ms
100	2^{12} * T_{WDT}	(3/18/130/1026) * T_{WDT}	407 ms
101	2^{14} * T_{WDT}	(3/18/130/1026) * T_{WDT}	1.638 s
110	2^{16} * T_{WDT}	(3/18/130/1026) * T_{WDT}	6.553 s
111	2^{18} * T_{WDT}	(3/18/130/1026) * T_{WDT}	26.214 s

Table 6-13 Watchdog Time-out Interval Selection



- *T_{WDT} : Watchdog Engine Clock Time Period*
- *T_{TIS} : Watchdog Timeout Interval Selection Period*
- *T_{RSTD}: Watchdog Reset Delay Period (Selectable 3/18/130/1026 * T_{WDT} delay period controlled by WTRDSEL)*
- *T_{RST} : Watchdog Reset Period*
- *T_{WTR} : Watchdog Reset Interval Period*
- *WTRDSEL : WDT_CTL[9:8]*

Figure 6-48 Watchdog Timing of Interrupt and Reset Signal

6.11.5 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
WDT Base Address: **WDT_BA = 0x4000_4000**				
WDT_CTL	WDT_BA+0x00	R/W	Watchdog Timer Control Register	0x0000_0070
WDT_IER	WDT_BA+0x04	R/W	Watchdog Timer Interrupt Enable Register	0x0000_0000
WDT_ISR	WDT_BA+0x08	R/W	Watchdog Timer Interrupt Status Register	0x0000_0000

6.11.6 Register Description

Watchdog Timer Control Register (WDT_CTL)

Register	Offset	R/W	Description	Reset Value
WDT_CTL	WDT_BA+0x00	R/W	Watchdog Timer Control Register	0x0000_0070

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th></tr>
<tr><td colspan="6">Reserved</td><td colspan="2">WTRDSEL</td></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>Reserved</td><td colspan="3">WTIS</td><td>WTE</td><td>WTWKE</td><td>WTRE</td><td>WTR</td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:10]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
<tr><td>[9:8]</td><td><b>WTRDSEL</b></td><td><b>Watchdog Timer Reset Delay Select</b>
When watchdog timeout happened, software has a time named watchdog reset delay period to clear watchdog timer to prevent watchdog reset happened. Software can select a suitable value of watchdog reset delay period for different watchdog timeout period.
<table>
<tr><th>WTRDSEL[1:0]</th><th>Description</th></tr>
<tr><td>00</td><td>Watchdog reset delay period is 1026 watchdog clock</td></tr>
<tr><td>01</td><td>Watchdog reset delay period is 130 watchdog clock</td></tr>
<tr><td>10</td><td>Watchdog reset delay period is 18 watchdog clock</td></tr>
<tr><td>11</td><td>Watchdog reset delay period is 3 watchdog clock</td></tr>
</table>
<b>Note:</b> This bit will be reset if watchdog reset happened</td></tr>
<tr><td>[7]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
<tr><td>[6:4]</td><td><b>WTIS</b></td><td><b>Watchdog Timer Interval Selection (Write Protect)</b>
Please refer to open lock sequence to program it.
The three bits select the time-out interval for the Watchdog timer. This count is free running counter.
Please refer to Table 6-13.</td></tr>
<tr><td>[3]</td><td><b>WTE</b></td><td><b>Watchdog Timer Enable Control (Write Protect)</b>
Please refer to open lock sequence to program it.
0 = Watchdog timer Disabled (this action will reset the internal counter).
1 = Watchdog timer Enabled.</td></tr>
<tr><td>[2]</td><td><b>WTWKE</b></td><td><b>Watchdog Timer Wake-up Function Enable Control (Write Protect)</b>
Please refer to open lock sequence to program it.</td></tr>
</table>

		0 = Watchdog timer Wake-up CPU function Disabled. 1 = Wake-up function Enabled so that Watchdog timer time-out can wake up CPU from Power-down mode.
[1]	**WTRE**	**Watchdog Timer Reset Function Enable Control (Write Protect)** Please refer to open lock sequence to program it. Setting this bit will enable the Watchdog timer reset function. 0 = Watchdog timer reset function Disabled. 1 = Watchdog timer reset function Enabled.
[0]	**WTR**	**Clear Watchdog Timer (Write Protect)** Please refer to open lock sequence to program it. Setting this bit will clear the Watchdog timer. 0 = No effect. 1 = Reset the contents of the Watchdog timer. **Note:** This bit will be auto cleared after 1 PCLK clock cycle.

Watchdog Timer Interrupt Enable Register (WDT_IER)

Register	Offset	R/W	Description	Reset Value
WDT_IER	WDT_BA+0x04	R/W	Watchdog Timer Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**WDT_IE**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**WDT_IE**	**Watchdog Timer Interrupt Enable Control** 0 = Watchdog timer interrupt Disabled. 1 = Watchdog timer interrupt Enabled.

Watchdog Timer Interrupt Status Register (WDT_ISR)

Register	Offset	R/W	Description	Reset Value
WDT_ISR	WDT_BA+0x08	R/W	Watchdog Timer Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**WDT_WAKE_IS**	**WDT_RST_IS**	**WDT_IS**

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**WDT_WAKE_IS**	**Watchdog Timer Wake-up Status** If Watchdog timer causes system to wake up from Power-down mode, this bit will be set to 1. It must be cleared by software with a write “1” to this bit. 0 = Watchdog timer does not cause system wake-up. 1 = Wake system up from Power-down mode by Watchdog time-out. **Note1:** When system in Power-down mode and watchdog time-out, hardware will set WDT_WAKE_IS and WDT_IS. **Note2:** After one engine clock, this bit can be cleared by writing “1” to it
[1]	**WDT_RST_IS**	**Watchdog Timer Reset Status** When the Watchdog timer initiates a reset, the hardware will set this bit. This flag can be read by software to determine the source of reset. Software is responsible to clear it manually by writing “1” to it. If WTRE is disabled, then the Watchdog timer has no effect on this bit. 0 = Watchdog timer reset did not occur. 1 = Watchdog timer reset occurs. **Note:** This bit is read only, but can be cleared by writing “1” to it.
[0]	**WDT_IS**	**Watchdog Timer Interrupt Status** If the Watchdog timer interrupt is enabled, then the hardware will set this bit to indicate that the Watchdog timer interrupt has occurred. If the Watchdog timer interrupt is not enabled, then this bit indicates that a time-out period has elapsed. 0 = Watchdog timer interrupt did not occur. 1 = Watchdog timer interrupt occurs. **Note:** This bit is read only, but can be cleared by writing “1” to it.

6.12 Window Watchdog Timer Controller

6.12.1 Overview

The purpose of Window Watchdog Timer is to perform a system reset within a specified window period to prevent software run to uncontrollable status by any unpredictable condition.

6.12.2 Features

- 6-bit down counter and 6-bit compare value to make the window period flexible
- Selectable WWDT clock pre-scale counter to make WWDT time-out interval variable

6.12.3 Block Diagram

The Window Watchdog Timer block diagram is shown as follows.

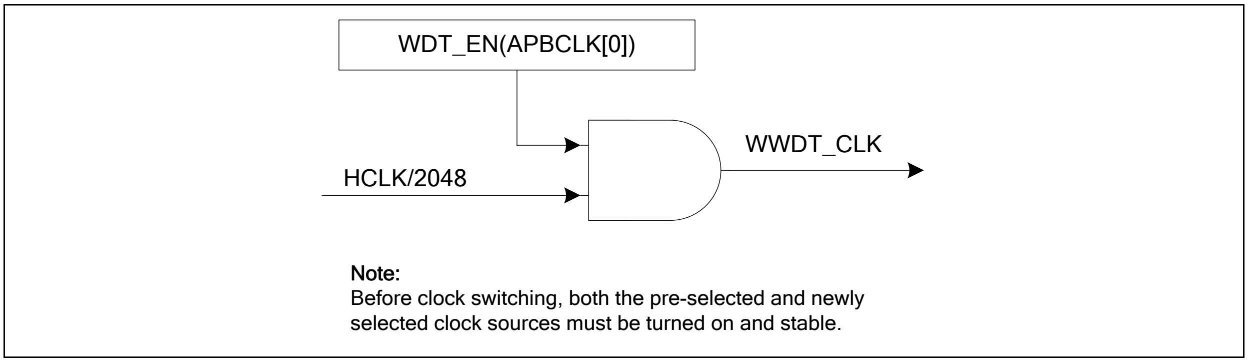

WDT_EN(APBCLK[0])
HCLK/2048
WWDT_CLK
Note:
Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.

Figure 6‑49 Window Watchdog Controller Block Diagram

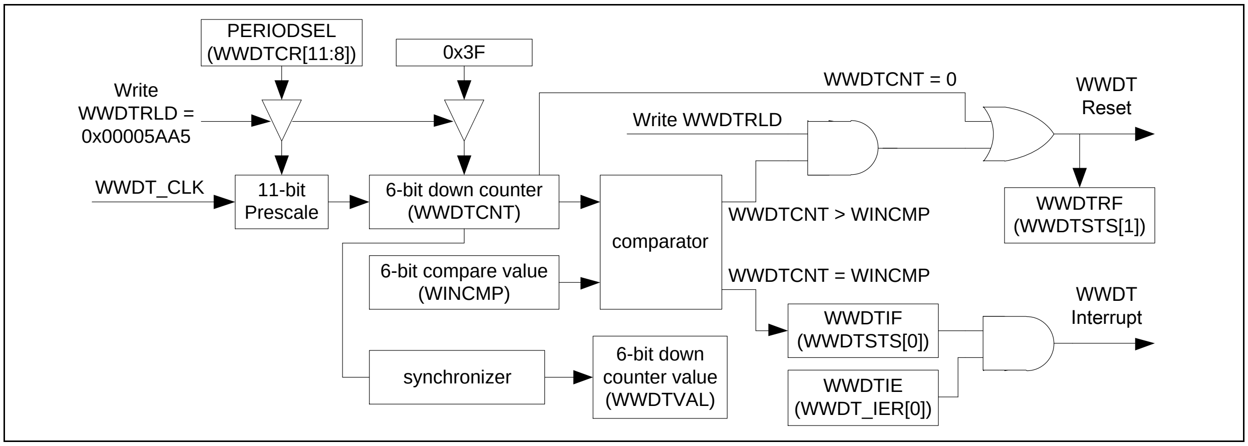


Figure 6‑50 Watchdog Controller Block Diagram

6.12.4 Functional Description

The window watchdog timer includes a 6-bit down counter with programmable prescaler to define different time-out intervals.

The clock source of 6-bit window watchdog timer is based on system clock divide 2048 with a programmable 11-bit prescaler. The programmable 11-bit prescaler is controlled by register PERIODSEL (WWDTCR[11:8]) and the correlate of PERIODSEL and prescaler value is list in following table.

PERIODSEL	Prescaler Value	Timeout Period	Timeout Interval 12 MHz/2048 = 5.859 KHz WWDT_CLK=5.859 KHz
0000	1	1 * 64 * T_{WWDT}	10.9 ms

0001	2	2 * 64 * T_{WWDT}	21.8 ms
0010	4	4 * 64 * T_{WWDT}	43.7 ms
0011	8	8 * 64 * T_{WWDT}	87.4 ms
0100	16	16 * 64 * T_{WWDT}	174.7 ms
0101	32	32 * 64 * T_{WWDT}	349.5 ms
0110	64	64 * 64 * T_{WWDT}	699.1 ms
0111	128	128 * 64 * T_{WWDT}	1.3981 s
1000	192	192 * 64 * T_{WWDT}	2.0971 s
1001	256	256 * 64 * T_{WWDT}	2.7962 s
1010	384	384 * 64 * T_{WWDT}	4.1943 s
1011	512	512 * 64 * T_{WWDT}	5.5924 s
1100	768	768 * 64 * T_{WWDT}	8.3886 s
1101	1024	1024 * 64 * T_{WWDT}	11.1848 s
1110	1536	1536 * 64 * T_{WWDT}	16.7772 s
1111	2048	2048 * 64 * T_{WWDT}	22.3696 s

Table 6‑14 Window Watchdog Prescaler Value Selection

The window watchdog timer can be enabled by software setting WWDTEN (WWDTCR[0]) to 1. As window watchdog timer is enabled, the down counter will start counting from 0x3F and cannot be stopped by software.

During WWDT down counting, the WWDT interrupt will happen if the counter value is equal to window watchdog timer compare value WINCMP (WWDTCR[21:16]] and WWDTIE(WWDT_IER[0]) is set to 1. The WWDT reset will happen if the WWDT counter value reaches to 0. Before WWDT counter down to 0, software can write certain value (0x00005AA5) to register WWDTRLD to reload 0x3F to WWDT counter to prevent WWDT reset happen and this reload action only active when WWDT counter value is equal or smaller than WINCMP. If software writes WWDTRLD during the period that WWDT counter larger than WINCMP, additional WWDT reset will happen to cause chip be reset.

When software writes certain value (0x00005AA5) to register WWDTRLD to reload WWDT counter, it need 3 window watchdog clocks to sync reload command to actually perform reload action. It means if software set window watchdog clock prescaler as divide 1, the compare value WINCMP (WWDTCR[21:16]) should larger than 2 or software will not able to reload WWDT counter before WWDT reset happened.

To prevent program run to unexpected code to disable window watchdog, the control register WWDTCR and WWDT_IER can only be written one time after chip power on or reset. In other words, once configured and enabled, window watchdog timer cannot be disabled or re-configure unless chip is reset. Window watchdog timer will stop counting while CPU is in idle mode or Power-down mode, and resume after CPU waked up.

6.12.5 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
WWDT Base Address: **WWDT_BA = 0x4000_4100**				
WWDTRLD	WWDT_BA+0x00	W	Window Watchdog Timer Reload Counter Register	0x0000_0000
WWDTCR	WWDT_BA+0x04	R/W	Window Watchdog Timer Control Register	0x003F_0800
WWDT_IER	WWDT_BA+0x08	R/W	Window Watchdog Timer Interrupt Enable Register	0x0000_0000
WWDTSTS	WWDT_BA+0x0C	R/W	Window Watchdog Timer Status Register	0x0000_0000
WWDTVAL	WWDT_BA+0x10	R	Window Watchdog Timer Counter Value Register	0x0000_003F

6.12.6 Register Description

Window Watchdog Timer Reload Counter Register (WWDTRLD)

Register	Offset	R/W	Description	Reset Value
WWDTRLD	WWDT_BA+0x00	W	Window Watchdog Timer Reload Counter Register	0x0000_0000

31	30	29	28	27	26	25	24
WWDTRLD							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WWDTRLD							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
WWDTRLD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WWDTRLD							

Bits	Description	
[31:0]	**WWDTRLD**	**Window Watchdog Timer Reload Counter Register** Writing 0x00005AA5 to this register will reload the Window Watchdog Timer counter value to 0x3F. **Note:** This register can only be written when WWDT counter value between 0 and WINCMP, otherwise WWDT will generate RESET signal.

Window Watchdog Timer Control Register (WWDTCR)

Register	Offset	R/W	Description	Reset Value
WWDTCR	WWDT_BA+0x04	R/W	Window Watchdog Timer Control Register	0x003F_0800

Note: This register can be write only one time after chip power on or reset.

31	30	29	28	27	26	25	24
DBGEN	Reserved						
23	22	21	20	19	18	17	16
Reserved		WINCMP					
15	14	13	12	11	10	9	8
Reserved				PERIODSEL			
7	6	5	4	3	2	1	0
Reserved							WWDTEN

Bits	Description	
[31]	**DBGEN**	**WWDT Debug Enable Control** 0 = WWDT stopped count if system is in Debug mode. 1 = WWDT still counted even system is in Debug mode.
[30:22]	**Reserved**	Reserved.
[21:16]	**WINCMP**	**WWDT Window Compare Register** Set this register to adjust the valid reload window. **Note:** WWDTRLD register can only be written when WWDT counter value between 0 and WINCMP, otherwise WWDT will generate RESET signal.
[15:12]	**Reserved**	Reserved.
[11:8]	**PERIODSEL**	**WWDT Pre-scale Period Select** These three bits select the pre-scale for the WWDT counter period. Please refer to Table 6‑14.
[7:1]	**Reserved**	Reserved.
[0]	**WWDTEN**	**Window Watchdog Enable Control** Set this bit to enable Window Watchdog timer. 0 = Window Watchdog timer function Disabled. 1 = Window Watchdog timer function Enabled.

Window Watchdog Timer Interrupt Enable Register (WWDT_IER)

Register	Offset	R/W	Description	Reset Value
WWDT_IER	WWDT_BA+0x08	R/W	Window Watchdog Timer Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**WWDTIE**

Note: This register can be write only one time after chip power on or reset.

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**WWDTIE**	**WWDT Interrupt Enable Control** Setting this bit will enable the Window Watchdog timer interrupt function. 0 = Watchdog timer interrupt function Disabled. 1 = Watchdog timer interrupt function Enabled.

Window Watchdog Timer Status Register (WWDTSTS)

Register	Offset	R/W	Description	Reset Value
WWDTSTS	WWDT_BA+0x0C	R/W	Window Watchdog Timer Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**WWDTRF**	**WWDTIF**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**WWDTRF**	**WWDT Reset Flag** When WWDT counter down count to 0 or write WWDTRLD during WWDT counter larger than WINCMP, chip will be reset and this bit is set to 1. This bit can be cleared by writing one to it.
[0]	**WWDTIF**	**WWDT Compare Match Interrupt Flag** When WWCMP match the WWDT counter, then this bit is set to 1. This bit can be cleared by writing one to it.

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

Window Watchdog Timer Counter Value Register (WWDTVAL)

Register	Offset	R/W	Description	Reset Value
WWDTVAL	WWDT_BA+0x10	R	Window Watchdog Timer Counter Value Register	0x0000_003F

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**WWDTVAL**					

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:0]	**WWDTVAL**	**WWDT Counter Value** This register reflects the current counter value of window watchdog.

6.13 RTC

6.13.1 Overview

Real Time Clock (RTC) unit provides user the real time and calendar message. The Clock Source (LXT) of RTC is from an external 32.768 kHz crystal connected at pins X32I and X32O (reference to pin Description) or from an external 32.768 kHz oscillator output fed at pin X32I. The RTC unit provides the time message (second, minute, hour) in Time Loading Register (TLR) as well as calendar message (day, month, year) in Calendar Loading Register (CLR). The data message is expressed in BCD format. This unit offers alarm function that user can preset the alarm time in Time Alarm Register (TAR) and alarm calendar in Calendar Alarm Register (CAR).

The RTC unit supports periodic Time Tick and Alarm Match interrupts. The periodic interrupt has 8 period options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second which are selected by TTR (RTC_TTR[2:0]). When RTC counter in TLR and CLR is equal to alarm setting time registers TAR and CAR, the alarm interrupt status (AIS (RTC_RIIR[0])) is set and the alarm interrupt is requested if the alarm interrupt is enabled (AIER (RTC_RIER[0])=1). The RTC Time Tick (if wake-up CPU function is enabled, (TWKE (RTC_TTR[3]) high) and Alarm Match can cause CPU wake-up from idle or Power-down mode.

6.13.2 Features

- One time counter (second, minute, hour) and calendar counter (day, month, year) for user to check the time
- Alarm register (second, minute, hour, day, month, year)
- 12-hour or 24-hour mode is selectable
- Leap year compensation automatically
- Day of week counter
- Frequency compensate register (FCR)
- All time and calendar message is expressed in BCD code
- Supports periodic time tick interrupt with 8 period options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second
- Supports 1, 2, 4, 8 and 16 seconds clock output (CLK_Hz) for frequency measuring
- Supports RTC Time Tick and Alarm Match interrupt
- Supports wake-up CPU from Power-down mode
- Supports 80 bytes spare registers and a snoop pin to clear the content of these spare registers

6.13.3 Block Diagram

The block diagram of Real Time Clock is depicted as follows.

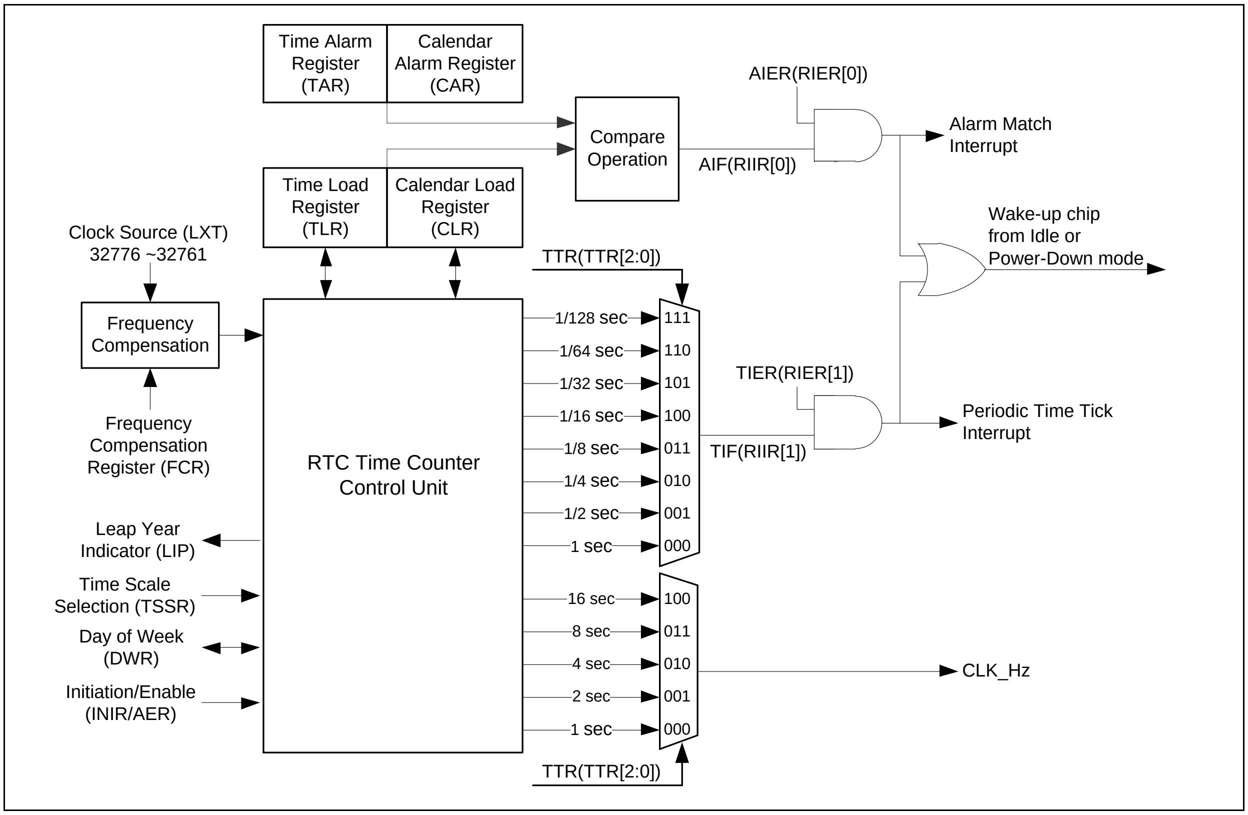

Time Alarm Register (TAR)
Calendar Alarm Register (CAR)
AIER(RIER[0])
Compare Operation
AIF(RIIR[0])
Alarm Match Interrupt
Time Load Register (TLR)
Calendar Load Register (CLR)
Clock Source (LXT) 32776 ~32761
Wake-up chip from Idle or Power-Down mode
TTR(TTR[2:0])
Frequency Compensation
1/128 sec
1/64 sec
1/32 sec
1/16 sec
1/8 sec
1/4 sec
1/2 sec
1 sec
111
110
101
100
011
010
001
000
TIER(RIER[1])
Frequency Compensation Register (FCR)
TIF(RIIR[1])
Periodic Time Tick Interrupt
RTC Time Counter Control Unit
Leap Year Indicator (LIP)
Time Scale Selection (TSSR)
Day of Week (DWR)
Initiation/Enable (INIR/AER)
16 sec
8 sec
4 sec
2 sec
1 sec
100
011
010
001
000
CLK_Hz
TTR(TTR[2:0])

Figure 6-51 RTC Core Block Diagram

6.13.4 Functional Description

6.13.4.1 Access to RTC register

Due to clock difference between RTC clock and system clock, when user writes new data to any one of the registers, the register will not be updated until 2 RTC clocks later (60us).

In addition, user must be aware that RTC block does not check whether loaded data is out of bounds or not. RTC does not check rationality between DWR and CLR either.

6.13.4.2 RTC Initiation

When RTC block is powered on, user has to write a number (0xa5eb1357) to RTC_INIR to reset all logic. RTC_INIR acts as hardware reset circuit. Once RTC_INIR has been set as 0xa5eb1357, there is no action for RTC if any value is programmed into RTC_INIR register.

6.13.4.3 RTC Read/Write Enable

Register AER bit 15~0 is served as RTC read/write password to protect RTC registers. AER (RTC_AER[15:0]) has to be set as 0xa965 to enable access restriction. Once it is set, it will take effect at least 512 RTC clocks (about 15ms). Programmer can read RTC enabled status flag (ENF (RTC_AER[16])) to check whether if RTC unit start operating or not.

6.13.4.4 Frequency Compensation

The RTC_FCR register allows software to make digital compensation to a clock input. Please follow the example and formula below to write the actual frequency of 32k crystal to RTC_FCR register. Following are the compensation examples for higher or lower than 32768 Hz.

Example 1:

Frequency counter measurement: 32773.65 Hz (> 32768 Hz)

FCR = (32768 * 0x200000) / 32773.65 = 0x1FFE96

Example 2:

Frequency counter measurement: 32763.25 Hz (< 32768 Hz)

FCR = (32768 * 0x200000) / 32763.25 = 0x200130

Note: The value of RTC_FCR register will be as default value (0x0020_0000) while the compensation is not executed. User can utilize a frequency counter to measure RTC clock source via clock output function in manufacturing. In the meanwhile, user can use clock output function to check the result of RTC frequency compensation.

6.13.4.5 Time and Calendar counter

RTC_TLR and RTC_CLR registers are used to load the time and calendar. RTC_TAR and RTC_CAR registers are used for alarm. They are all represented by BCD.

6.13.4.6 12/24 hour Time Scale Selection

The 12/24 hour time scale selection depends on 24hr_12hr (RTC_TSSR[0]).

6.13.4.7 Day of the week counter

The RTC unit provides day of week in Day of the Week Register (RTC_DWR). The value is defined from 0 to 6 to represent Sunday to Saturday respectively.

6.13.4.8 Periodic Time Tick Interrupt

The periodic interrupt has 8 period option 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second which are selected by TTR(TTR[2:0]). When periodic time tick interrupt is enabled by setting TIER (RTC_RIER[1]) to 1, the Periodic Time Tick Interrupt is requested periodically in the period selected by TTR (RTC_TTR[2:0]).

6.13.4.9 Hz Clock output

RTC support output a Hz clock (CLK_Hz) for frequency measuring.

The clock output frequency has 5 period option 1, 2, 4, 8 and 16 seconds which are selected by TTR(RTC_TTR[2:0]).

6.13.4.10 Alarm Interrupt

When RTC counter in TLR and CLR is equal to alarm setting time TAR and CAR the alarm interrupt status (AIS (RIIR[0])) is set and the alarm interrupt is requested if the alarm interrupt is enabled (AIER (RIER[0])=1).

Application Note:

TAR, CAR, TLR and CLR registers are all BCD counter.

Programmer has to make sure that the loaded values are reasonable. For example, Load CLR as 201a (year), 13 (month), 00 (day), or CLR does not match with DWR, etc.

Reset state:

Register	**Reset State**
AER	0
CLR	05/1/1 (year/month/day)
TLR	00:00:00 (hour : minute : second)
CAR	00/00/00 (year/month/day)
TAR	00:00:00 (hour : minute : second)
TSSR	1 (24 hr mode)
DWR	6 (Saturday)
RIER	0
RIIR	0
LIR	0
TTR	0

In CLR and CAR, only 2 BCD digits are used to express “year”. It is assumed that 2 BCD digits of xy denote 20xy, but not 19xy or 21xy.

6.13.4.11 Spare registers and snoop pin

The RTC is equipped with 80 bytes spare registers to store important user information, and also has a snoop function to detect the transition of snoop pin. Once the transition defined in SNOOPEDGE (RTC_SPRCTL[1]) is detected in snoop pin and SNOOPEN (RTC_SPRCTL[0]) is 1, the 80 bytes spare registers will be cleared by RTC automatically.

6.13.5 Register Map

R: read only, **W:** write only, **R/W:** both read and write, **C:** write 1 to clear

Register	Offset	R/W	Description	Reset Value
RTC Base Address: **RTC_BA = 0x4000_8000**				
RTC_INIR	RTC_BA+0x00	R/W	RTC Initiation Register	0x0000_0000
RTC_AER	RTC_BA+0x04	R/W	RTC Access Enable Register	0x0000_0000
RTC_FCR	RTC_BA+0x08	R/W	RTC Frequency Compensation Register	0x0020_0000
RTC_TLR	RTC_BA+0x0C	R/W	Time Loading Register	0x0000_0000
RTC_CLR	RTC_BA+0x10	R/W	Calendar Loading Register	0x0005_0101
RTC_TSSR	RTC_BA+0x14	R/W	Time Scale Selection Register	0x0000_0001
RTC_DWR	RTC_BA+0x18	R/W	Day of the Week Register	0x0000_0006
RTC_TAR	RTC_BA+0x1C	R/W	Time Alarm Register	0x0000_0000
RTC_CAR	RTC_BA+0x20	R/W	Calendar Alarm Register	0x0000_0000
RTC_LIR	RTC_BA+0x24	R	Leap Year Indicator Register	0x0000_0000
RTC_RIER	RTC_BA+0x28	R/W	RTC Interrupt Enable Register	0x0000_0000
RTC_RIIR	RTC_BA+0x2C	R/W	RTC Interrupt Indication Register	0x0000_0000
RTC_TTR	RTC_BA+0x30	R/W	RTC Time Tick Register	0x0000_0000
RTC_SPRCTL	RTC_BA+0x3C	R/W	RTC Spare Functional Control Register	0x0000_0080
RTC_SPR0	RTC_BA+0x40	R/W	RTC Spare Register 0	0x0000_0000
RTC_SPR1	RTC_BA+0x44	R/W	RTC Spare Register 1	0x0000_0000
RTC_SPR2	RTC_BA+0x48	R/W	RTC Spare Register 2	0x0000_0000
RTC_SPR3	RTC_BA+0x4C	R/W	RTC Spare Register 3	0x0000_0000
RTC_SPR4	RTC_BA+0x50	R/W	RTC Spare Register 4	0x0000_0000
RTC_SPR5	RTC_BA+0x54	R/W	RTC Spare Register 5	0x0000_0000
RTC_SPR6	RTC_BA+0x58	R/W	RTC Spare Register 6	0x0000_0000
RTC_SPR7	RTC_BA+0x5C	R/W	RTC Spare Register 7	0x0000_0000
RTC_SPR8	RTC_BA+0x60	R/W	RTC Spare Register 8	0x0000_0000
RTC_SPR9	RTC_BA+0x64	R/W	RTC Spare Register 9	0x0000_0000
RTC_SPR10	RTC_BA+0x68	R/W	RTC Spare Register 10	0x0000_0000
RTC_SPR11	RTC_BA+0x6C	R/W	RTC Spare Register 11	0x0000_0000
RTC_SPR12	RTC_BA+0x70	R/W	RTC Spare Register 12	0x0000_0000

Register	Offset	R/W	Description	Reset Value
RTC_SPR13	RTC_BA+0x74	R/W	RTC Spare Register 13	0x0000_0000
RTC_SPR14	RTC_BA+0x78	R/W	RTC Spare Register 14	0x0000_0000
RTC_SPR15	RTC_BA+0x7C	R/W	RTC Spare Register 15	0x0000_0000
RTC_SPR16	RTC_BA+0x80	R/W	RTC Spare Register 16	0x0000_0000
RTC_SPR17	RTC_BA+0x84	R/W	RTC Spare Register 17	0x0000_0000
RTC_SPR18	RTC_BA+0x88	R/W	RTC Spare Register 18	0x0000_0000
RTC_SPR19	RTC_BA+0x8C	R/W	RTC Spare Register 19	0x0000_0000

6.13.6 Register Description

RTC Initiation Register (RTC_INIR)

Register	Offset	R/W	Description	Reset Value
RTC_INIR	RTC_BA+0x00	R/W	RTC Initiation Register	0x0000_0000

31	30	29	28	27	26	25	24
INIR							
23	22	21	20	19	18	17	16
INIR							
15	14	13	12	11	10	9	8
INIR							
7	6	5	4	3	2	1	0
INIR							ACTIVE

Bits	Description	
[31:1]	**INIR**	**RTC Initiation (Write Only)** When RTC block is powered on, RTC is at reset state. User has to write a number (0x a5eb1357) to INIR to make RTC leaving reset state. Once the INIR is written as 0xa5eb1357, the RTC will be in un-reset state permanently. The INIR is a write-only field and read value will be always “0”.
[0]	**ACTIVE**	**RTC Active Status (Read Only)** 0 = RTC is at reset state. 1 = RTC is at normal active state.

RTC Access Enable Register (RTC_AER)

Register	Offset	R/W	Description	Reset Value
RTC_AER	RTC_BA+0x04	R/W	RTC Access Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**ENF**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
AER							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AER							

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:17]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
<tr><td>[16]</td><td><b>ENF</b></td><td><b>RTC Register Access Enable Flag (Read Only)</b>
0 = RTC register read/write Disabled.
1 = RTC register read/write Enabled.
This bit will be set after AER[15:0] register is load a 0xA965, and be cleared automatically 512 RTC clocks or AER[15:0] is not 0xA965.
<table>
<tr><th>ENF
Register</th><th>1</th><th>0</th></tr>
<tr><td>INIR</td><td>R/W</td><td>R/W</td></tr>
<tr><td>AER</td><td>R/W</td><td>R/W</td></tr>
<tr><td>FCR</td><td>R/W</td><td>-</td></tr>
<tr><td>TLR</td><td>R/W</td><td>R</td></tr>
<tr><td>CLR</td><td>R/W</td><td>R</td></tr>
<tr><td>TSSR</td><td>R/W</td><td>R/W</td></tr>
<tr><td>DWR</td><td>R/W</td><td>R</td></tr>
<tr><td>TAR</td><td>R/W</td><td>-</td></tr>
<tr><td>CAR</td><td>R/W</td><td>-</td></tr>
<tr><td>LIR</td><td>R</td><td>R</td></tr>
<tr><td>RIER</td><td>R/W</td><td>R/W</td></tr>
<tr><td>RIIR</td><td>R/W</td><td>R/W</td></tr>
<tr><td>TTR</td><td>R/W</td><td>-</td></tr>
<tr><td>SPRCTL</td><td>R/W</td><td>-</td></tr>
<tr><td>SPR0~SPR19</td><td>R/W</td><td>-</td></tr>
</table></td></tr>
</table>

[15:0]	**AER**	**RTC Register Access Enable Password (Write Only)** 0xa965 = RTC access Enable.d Others = RTC access Disable.d

RTC Frequency Compensation Register (RTC_FCR)

Register	Offset	R/W	Description	Reset Value
RTC_FCR	RTC_BA+0x08	R/W	RTC Frequency Compensation Register	0x0020_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**FCR**					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FCR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FCR							

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:0]	**FCR**	**Frequence Compensation Register** FCR = 32768 * 0x200000 / (LXT period). LXT period: the clock period (Hz) of LXT.

RTC Time Loading Register (RTC_TLR)

Register	Offset	R/W	Description	Reset Value
RTC_TLR	RTC_BA+0x0C	R/W	Time Loading Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		10HR		1HR			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	10MIN			1MIN			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	10SEC			1SEC			

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:20]	**10HR**	10 Hour Time Digit (0~2)
[19:16]	**1HR**	1 Hour Time Digit (0~9)
[15]	**Reserved**	Reserved.
[14:12]	**10MIN**	10 Min Time Digit (0~5)
[11:8]	**1MIN**	1 Min Time Digit (0~9)
[7]	**Reserved**	Reserved.
[6:4]	**10SEC**	10 Sec Time Digit (0~5)
[3:0]	**1SEC**	1 Sec Time Digit (0~9)

Note: TLR is a BCD digit counter and RTC will not check the loaded data.

The reasonable value range is listed in the parenthesis.

RTC Calendar Loading Register (RTC_CLR)

Register	Offset	R/W	Description	Reset Value
RTC_CLR	RTC_BA+0x10	R/W	Calendar Loading Register	0x0005_0101

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="4">10YEAR</td><td colspan="4">1YEAR</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th></tr>
<tr><td colspan="3">Reserved</td><td>10MON</td><td colspan="4">1MON</td></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="2">Reserved</td><td colspan="2">10DAY</td><td colspan="4">1DAY</td></tr>
</table>

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:20]	**10YEAR**	10 Year Calendar Digit (0~9)
[19:16]	**1YEAR**	1 Year Calendar Digit (0~9)
[12]	**10MON**	10 Month Calendar Digit (0~1)
[11:8]	**1MON**	1 Month Calendar Digit (0~9)
[5:4]	**10DAY**	10 Day Calendar Digit (0~3)
[3:0]	**1DAY**	1 Day Calendar Digit (0~9)

Note: CLR is a BCD digit counter and RTC will not check loaded data.

The reasonable value range is listed in the parenthesis.

RTC Time Scale Selection Register (RTC_TSSR)

Register	Offset	R/W	Description	Reset Value
RTC_TSSR	RTC_BA+0x14	R/W	Time Scale Selection Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**24hr_12hr**

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:1]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
<tr><td>[0]</td><td><b>24hr_12hr</b></td><td><b>24-hour / 12-hour Mode Selection</b>
It indicates that TLR and TAR are in 24-hour mode or 12-hour mode
0 = select 12-hour time scale with AM and PM indication.
1 = select 24-hour time scale.
<table>
<tr><th>24-hour Time Scale</th><th>12-hour Time Scale</th><th>24-hour Time Scale</th><th>12-hour Time Scale (PM Time + 20)</th></tr>
<tr><td>00</td><td>12(AM12)</td><td>12</td><td>32(PM12)</td></tr>
<tr><td>01</td><td>01(AM01)</td><td>13</td><td>21(PM01)</td></tr>
<tr><td>02</td><td>02(AM02)</td><td>14</td><td>22(PM02)</td></tr>
<tr><td>03</td><td>03(AM03)</td><td>15</td><td>23(PM03)</td></tr>
<tr><td>04</td><td>04(AM04)</td><td>16</td><td>24(PM04)</td></tr>
<tr><td>05</td><td>05(AM</td><td>17</td><td>25(P</td></tr>
</table>
</td></tr>
</table>

	05)		M05)
06	06(AM06)	18	26(PM06)
07	07(AM07)	19	27(PM07)
08	08(AM08)	20	28(PM08)
09	09(AM09)	21	29(PM09)
10	10(AM10)	22	30(PM10)
11	11(AM11)	23	31(PM11)

RTC Day of the Week Register (RTC_DWR)

Register	Offset	R/W	Description	Reset Value
RTC_DWR	RTC_BA+0x18	R/W	Day of the Week Register	0x0000_0006

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**DWR**		

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**DWR**	**Day of the Week Register** 000 = Sunday. 001 = Monday. 010 = Tuesday. 011 = Wednesday. 100 = Thursday. 101 = Friday. 110 = Saturday.

RTC Time Alarm Register (RTC_TAR)

Register	Offset	R/W	Description	Reset Value
RTC_TAR	RTC_BA+0x1C	R/W	Time Alarm Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**10HR**		**1HR**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**10MIN**			**1MIN**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**10SEC**			**1SEC**			

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:20]	**10HR**	10 Hour Time Digit of Alarm Setting (0~2)
[19:16]	**1HR**	1 Hour Time Digit of Alarm Setting (0~9)
[15]	**Reserved**	Reserved.
[14:12]	**10MIN**	10 Min Time Digit of Alarm Setting (0~5)
[11:8]	**1MIN**	1 Min Time Digit of Alarm Setting (0~9)
[7]	**Reserved**	Reserved.
[6:4]	**10SEC**	10 Sec Time Digit of Alarm Setting (0~5)
[3:0]	**1SEC**	1 Sec Time Digit of Alarm Setting (0~9)

Note:

TAR is a BCD digit counter and RTC will not check loaded data.

The reasonable value range is listed in the parenthesis.

The register can be read back after the RTC unit is active by AER

RTC Calendar Alarm Register (RTC_CAR)

Register	Offset	R/W	Description	Reset Value
RTC_CAR	RTC_BA+0x20	R/W	Calendar Alarm Register	0x0000_0000

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="4">10YEAR</td><td colspan="4">1YEAR</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th></tr>
<tr><td colspan="3">Reserved</td><td>10MON</td><td colspan="4">1MON</td></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="2">Reserved</td><td colspan="2">10DAY</td><td colspan="4">1DAY</td></tr>
</table>

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:20]	**10YEAR**	10 Year Calendar Digit of Alarm Setting (0~9)
[19:16]	**1YEAR**	1 Year Calendar Digit of Alarm Setting (0~9)
[12]	**10MON**	10 Month Calendar Digit of Alarm Setting (0~1)
[11:8]	**1MON**	1 Month Calendar Digit of Alarm Setting (0~9)
[5:4]	**10DAY**	10 Day Calendar Digit of Alarm Setting (0~3)
[3:0]	**1DAY**	1 Day Calendar Digit of Alarm Setting (0~9)

Note:

CAR is a BCD digit counter and RTC will not check loaded data.

The reasonable value range is listed in the parenthesis.

The register can be read back after the RTC unit is active by AER

RTC Leap Year Indication Register (RTC_LIR)

Register	Offset	R/W	Description	Reset Value
RTC_LIR	RTC_BA+0x24	R	Leap Year Indicator Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**LIR**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**LIR**	**Leap Year Indication REGISTER (Read Only)** 0 = This year is not a leap year. 1 = This year is leap year.

RTC Interrupt Enable Register (RTC_RIER)

Register	Offset	R/W	Description	Reset Value
RTC_RIER	RTC_BA+0x28	R/W	RTC Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**SNOOPIER**	**TIER**	**AIER**

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**SNOOPIER**	**Snooper Pin Event Detection Interrupt Enable Control** 0 = Snooper Pin Event Detection Interrupt Disabled. 1 = Snooper Pin Event Detection Interrupt Enabled.
[1]	**TIER**	**Time Tick Interrupt and Wake-up by Tick Enable Control** 0 = RTC Time Tick Interrupt Disabled. 1 = RTC Time Tick Interrupt Enabled.
[0]	**AIER**	**Alarm Interrupt Enable Control** 0 = RTC Alarm Interrupt Disabled. 1 = RTC Alarm Interrupt Enabled.

RTC Interrupt Indication Register (RTC_RIIR)

Register	Offset	R/W	Description	Reset Value
RTC_RIIR	RTC_BA+0x2C	R/W	RTC Interrupt Indication Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**SNOOPIS**	**TIS**	**AIS**

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**SNOOPIS**	**Snooper Pin Event Detection Interrupt Status** When SNOOPEN is high and an event defined by SNOOPEDGE detected in snooper pin, this flag will be set. While this bit is set and SNOOPIER (RTC_RIER[2]) is also high, RTC will generate an interrupt to CPU. Write “1” to clear this bit to “0”. 0 = Snooper pin event defined by SNOOPEDGE (RTC_SPRCTL[1]) never detected. 1 = Snooper pin event defined by SNOOPEDGE (RTC_SPRCTL[1]) detected.
[1]	**TIS**	**RTC Time Tick Interrupt Status** RTC unit will set this bit to high periodically in the period selected by TTR (RTC_TTR[2:0]). When this bit is set and TIER (RTC_RIER[1]) is also high, RTC will generate an interrupt to CPU. This bit is cleared by writing “1” to it through software. 0 = RCT Time Tick Interrupt condition never occurred. 1 = RTC Time Tick Interrupt is requested.
[0]	**AIS**	**RTC Alarm Interrupt Status** RTC unit will set AIS to high once the RTC real time counters TLR and CLR reach the alarm setting time registers TAR and CAR. When this bit is set and AIER (RTC_RIER[0]) is also high, RTC will generate an interrupt to CPU. This bit is cleared by writing “1” to it through software. 0 = RCT Alarm Interrupt condition never occurred. 1 = RTC Alarm Interrupt is requested if AIER (RTC_RIER[0])=1.

RTC Time Tick Registeter (RTC_TTR)

Register	Offset	R/W	Description	Reset Value
RTC_TTR	RTC_BA+0x30	R/W	RTC Time Tick Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**TWKE**	**TTR**		

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:1]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
<tr><td>[3]</td><td><b>TWKE</b></td><td><b>RTC Timer Wake-up CPU Function Enable Control</b>
If TWKE is set before CPU enters Power-down mode, when a RTC Time Tick, CPU will be wakened up by RTC unit.
0 = Time Tick wake-up CPU function Disabled.
1 = Wake-up function Enabled so that CPU can be wake up from Power-down mode by Time Tick.
<b>Note:</b> Tick timer setting follows the TTR (RTC_TTR[2:0]) description.</td></tr>
<tr><td>[2:0]</td><td><b>TTR</b></td><td><b>Time Tick Register</b>
The RTC time tick period for Periodic Time Tick Interrupt request.
<table>
<tr><td>TTR[2:0]</td><td>Time tick (second)</td></tr>
<tr><td>000</td><td>1</td></tr>
<tr><td>001</td><td>1/2</td></tr>
<tr><td>010</td><td>1/4</td></tr>
<tr><td>011</td><td>1/8</td></tr>
<tr><td>100</td><td>1/16</td></tr>
<tr><td>101</td><td>1/32</td></tr>
<tr><td>110</td><td>1/64</td></tr>
<tr><td>111</td><td>1/128</td></tr>
</table>
This register also controls the clock_out frequency.
<table>
<tr><td>TTR[2:0]</td><td>clock_out frequency (second)</td></tr>
<tr><td>000</td><td>1</td></tr>
<tr><td>001</td><td>2</td></tr>
</table></td></tr>
</table>

		010	4
		011	8
		100	16
		101	reserved
		110	reserved
		111	reserved
		Note: This register can be read back after the RTC is active by AER.	

RTC Spare Function Controller Register (RTC_SPRCTL)

Register	Offset	R/W	Description	Reset Value
RTC_SPRCTL	RTC_BA+0x3C	R/W	RTC Spare Functional Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**SNOOPEDGE**	**SNOOPEN**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**SNOOPEDGE**	**Snooper Active Edge Selection** This bit defines which edge of snooper pin will generate a snooper pin detected event to clear the 20 spare registers. 0 = Rising edge of snooper pin generates snooper pin detected event. 1 = Falling edge of snooper pin generates snooper pin detected event.
[0]	**SNOOPEN**	**Snooper Pin Event Detection Enable Control** This bit enables the snooper pin event detection. When this bit is set high and an event defined by SNOOPEDGE (RTC_SPRCTL[1]) detected, the 20 spare registers will be cleared to “0” by hardware automatically. And, the SNOOPIS (RTC_RIIR[2]) will also be set. In addition, RTC will also generate wake-up event to wake system up. 0 = Snooper pin event detection function Disabled. 1 = Snooper pin event detection function Enabled.

RTC Spare Register X (RTC_SPRx)

Register	Offset	R/W	Description	Reset Value
RTC_SPR0	RTC_BA+0x40	R/W	RTC Spare Register 0	0x0000_0000
RTC_SPR1	RTC_BA+0x44	R/W	RTC Spare Register 1	0x0000_0000
RTC_SPR2	RTC_BA+0x48	R/W	RTC Spare Register 2	0x0000_0000
RTC_SPR3	RTC_BA+0x4C	R/W	RTC Spare Register 3	0x0000_0000
RTC_SPR4	RTC_BA+0x50	R/W	RTC Spare Register 4	0x0000_0000
RTC_SPR5	RTC_BA+0x54	R/W	RTC Spare Register 5	0x0000_0000
RTC_SPR6	RTC_BA+0x58	R/W	RTC Spare Register 6	0x0000_0000
RTC_SPR7	RTC_BA+0x5C	R/W	RTC Spare Register 7	0x0000_0000
RTC_SPR8	RTC_BA+0x60	R/W	RTC Spare Register 8	0x0000_0000
RTC_SPR9	RTC_BA+0x64	R/W	RTC Spare Register 9	0x0000_0000
RTC_SPR10	RTC_BA+0x68	R/W	RTC Spare Register 10	0x0000_0000
RTC_SPR11	RTC_BA+0x6C	R/W	RTC Spare Register 11	0x0000_0000
RTC_SPR12	RTC_BA+0x70	R/W	RTC Spare Register 12	0x0000_0000
RTC_SPR13	RTC_BA+0x74	R/W	RTC Spare Register 13	0x0000_0000
RTC_SPR14	RTC_BA+0x78	R/W	RTC Spare Register 14	0x0000_0000
RTC_SPR15	RTC_BA+0x7C	R/W	RTC Spare Register 15	0x0000_0000
RTC_SPR16	RTC_BA+0x80	R/W	RTC Spare Register 16	0x0000_0000
RTC_SPR17	RTC_BA+0x84	R/W	RTC Spare Register 17	0x0000_0000
RTC_SPR18	RTC_BA+0x88	R/W	RTC Spare Register 18	0x0000_0000
RTC_SPR19	RTC_BA+0x8C	R/W	RTC Spare Register 19	0x0000_0000

31	30	29	28	27	26	25	24
SPARE [31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SPARE [23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SPARE [15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPARE [7:0]							

Bits	Description	
[31:0]	**SPARE**	**SPARE** This field is used to store back-up information defined by software. This field will be cleared by hardware automatically once a snooper pin event is detected.

6.14 UART Controller

6.14.1 Overview

The UART Controller provides up to two channels of Universal Asynchronous Receiver/Transmitter (UART) modules and performs Normal Speed UART, and supports flow control function. The Universal Asynchronous Receiver/Transmitter (UART) performs a serial-to-parallel conversion on data received from the peripheral, and a parallel-to-serial conversion on data transmitted from the CPU.

The UART controller also supports IrDA (SIR), LIN Master/Slave and RS-485 function modes.

There are four conditions to wake-up the system and it also supports PWM channel source selection to modulate the PWM and the UART transmitter.

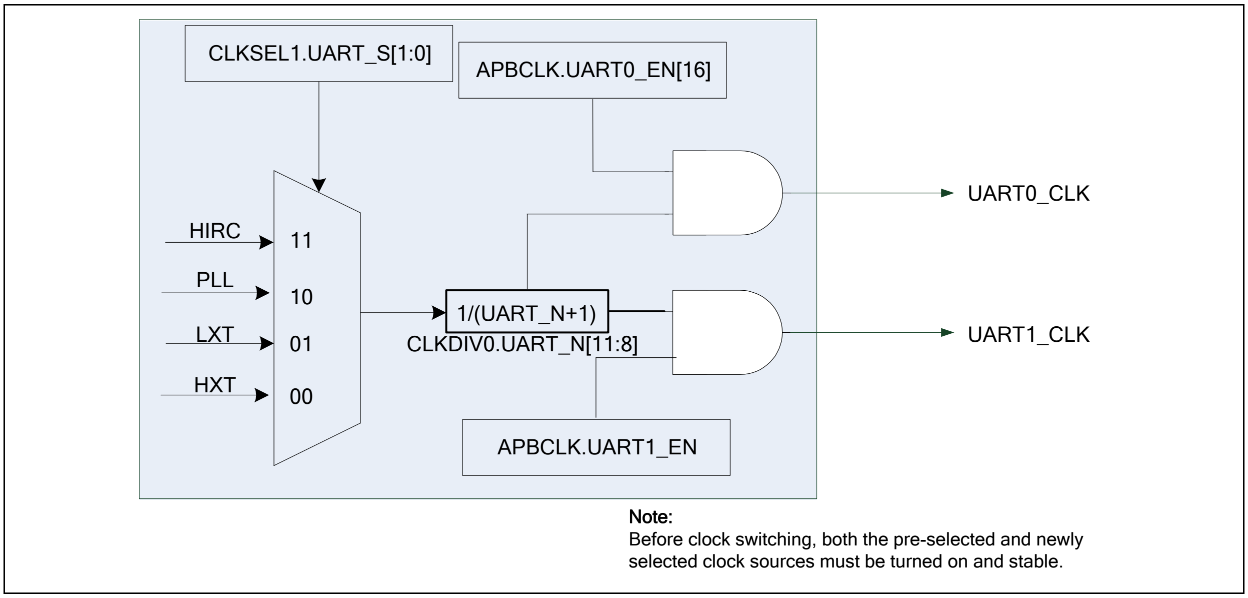
6.14.2 Features

- Full duplex, asynchronous communications.
- Separate receiving / transmitting 16 bytes entry FIFO for data payloads.
- Supports hardware auto-flow control/flow control function (CTSn, RTSn) and programmable (CTSn, RTSn) flow control trigger level.
- Supports programmable baud rate generator.
- Supports auto-baud rate detect and baud rate compensation function.
- Supports programmable receiver buffer trigger level.
- Supports incoming data or CTSn or received FIFO is equal to the RFITL or RS-485 AAD mode address matched to wake-up function.
- Supports 9 bit receiver buffer time-out detection function.
- All UART Controller can be served by the PDMA.
- Programmable transmitting data delay time between the last stop bit leaving the TX-FIFO and the de-assertion by setting DLY (UART_TMCTL[23:16]) register.
- Supports IrDA SIR function mode
- Supports LIN function mode.
- Supports RS-485 function mode.
- Supports PWM modulation

6.14.3 Block Diagram

The UART clock control and block diagram are shown as follows. The UART controller is completely asynchronous design with two clock domains, PCLK and engine clock.

Note: the PCLK should be higher than or equal to the frequency of peripheral clock (UARTx_CLK).



Note:
Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.

Figure 6-52 UART Clock Control Diagram

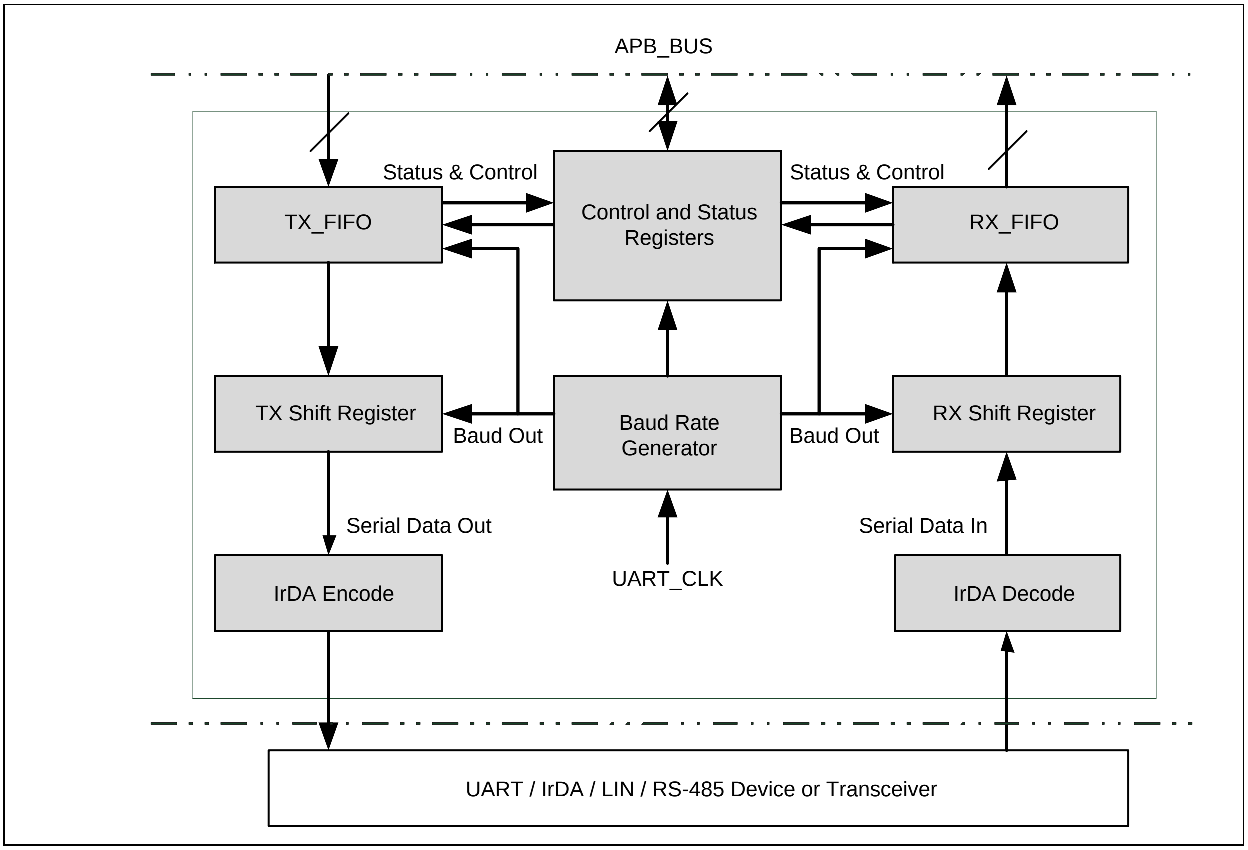


Figure 6-53 UART Block Diagram

TX_FIFO

The transmitter buffered is a 16 byte FIFO to reduce the number of interrupts presented to the CPU.

RX_FIFO

The receiver buffered is a 16 byte FIFO (plus three error bits per byte) to reduce the number of interrupts presented to the CPU.

TX Shift Register

The block shifts the transmitting data out serially control block.

RX Shift Register

The block shifts the receiving data in serially control block.

Baud Rate Generator

Divide the external clock or internal clock by the divisor to get the desired baud rate clock. Refer to for baud rate equation.

IrDA Encode

This block is the IrDA encode control block.

IrDA Decode

This block is the IrDA decode control block.

Control and Status Register

This is a register set, including the transfer line control registers (UART_TLCTL), transfer status registers (UART_TRSR), and control register (UART_CTL) for transmitter and receiver. The time-out control register (UART_TMCTL) identifies the condition of time-out interrupt. This register set also includes the interrupt enable register (UART_IER) and interrupt status register (UART_ISR) to enable or disable the responding interrupt and to identify the occurrence of the responding interrupt. There are nine types of interrupts including receiver threshold level reaching interrupt (INT_RDA), transmitter FIFO empty interrupt (INT_THRE), line status interrupt (break error, parity error, framing error or RS-485 interrupt) (INT_RLS), time-out interrupt (INT_TOUT), MODEM status interrupt (INT_MODEM), Buffer error interrupt (INT_BUF_ERR), wake-up interrupt (INT_WAKE), auto-baud rate detect or auto-baud rate counter overflow flag (INT_ABAUD) of LIN function interrupt (INT_LIN).

6.14.4 Basic Configuration

- The UART Controller function pins are configured in GPA_MFP, GPB_MFP, GPC_MFP and GPF_MFP registers for UART.
- The UART Controller clock are enabled in UART_EN(APBCLK[17:16]) for UART.
- The UART Controller clock source is selected by UART_S (CLKSEL[25:24]/CLKSEL1[1:0]).
- The UART Controller clock prescaler is determined by UART_N(CLKDIV0[11:8]).

6.14.5 Functional Description

The UART Controller supports four function modes including UART, IrDA, RS-485 and LIN function modes. User can select a function by setting the UA_FUN_SEL register. The UART baud rate is up to 1 Mbps.

6.14.5.1 Line Function Mode

The UART Controller supports fully programmable serial-interface characteristics by setting the UA_LCR register. Software can use the UA_LCR register to program the word length, stop bit and parity bit. The following tables list the UART word and stop bit length settings and the UART parity bit settings.

NSB (UA_LCR[2])	WLS (UA_LCR[1:0])	Word Length (Bit)	Stop Length (Bit)
0	00	5	1
0	01	6	1
0	10	7	1
0	11	8	1
1	00	5	1.5
1	01	6	2
1	10	7	2
1	11	8	2

Table 6‑15 UART Line Control of Word and Stop Length Setting

Parity Type	SPE (UA_LCR[5])	EPE (UA_LCR[4])	PBE (UA_LCR[3])	Description
No Parity	x	x	0	No parity bit output.
Odd Parity	0	0	1	Odd Parity is calculated by adding all the “1’s” in a data stream and adding a parity bit to the total bits, to make the total count an odd number.
Even Parity	0	1	1	Even Parity is calculated by adding all the “1’s” in a data stream and adding a parity bit to the total bits, to make the count an even number.
Forced Mask Parity	1	0	1	Parity bit always logic 1. Parity bit on the serial byte is set to “1” regardless of total number of “1’s” (even or odd counts).
Forced Space Parity	1	1	1	Parity bit always logic 0. Parity bit on the serial byte is set to “0” regardless of total number of “1’s” (even or odd counts).

Table 6‑16 UART Line Control of Parity Bit Setting

Note: User cannot change line controller setting when TE_F(UA_FSR[11]) is not empty

6.14.5.2 Baud Rate Generation

The UART Controller includes a programmable baud rate generator capable of dividing clock input by dividers to produce the serial clock that transmitter and receiver need. The baud rate equation is Baud Rate = UART_CLK / M * [BRD + 1], where M and BRD are defined in Baud Rate Divider Register (UART_BAUD). The following tables list the UART baud rate equations in the various conditions and UART baud rate parameter settings. There is no error for the baud rate results calculated through the baud rate parameter and register setting below. In IrDA function mode, the baud rate generator must be set in Mode 0.

DIV_16_EN	**Baud Rate Equation**
0 (Mode 0)	UART_CLK / (BRD+1), BRD must >2
1 (Mode 1)	UART_CLK / [16 * (BRD+1)]

Table 6‑17 UART Baud Rate Equation

System Clock =12 MHz		
Baud rate	Mode 0	Mode 1
921600	BRD=12	Not Supported
460800	BRD =25	Not Supported
230400	BRD =51	BRD =2
115200	BRD =103	BRD =6
57600	BRD =207	BRD =12
38400	BRD =311	BRD =19
19200	BRD =624	BRD =38
9600	BRD =1249	BRD =77
4800	BRD =2499	BRD =155

Table 6‑18 UART Baud Rate Setting

6.14.5.3 Baud Rate Compensation

The UART controller supports baud rate compensation function. It is used to optimize the precision in each bit. The precision of the compensation is half of UART module clock because there is BR_COM_DEC bit (UART_BR_COMP[31]), to define the positive or negative compensation in each bit. If the BR_COMP_DEC (UART_BR_COMP[31]) = 0, it is positive compensation for each bit, one more module clock will be append in the compensated bit. If the BR_COMP_DEC (UART_BR_COMP[31]) = 1, it is negedge compensation for each bit, decrease one module clock in the compensated bit.

There is 9-bits location, BR_COMP[8:0], can be configured by user to define the relative bit is compensated or not. BR_COMP[7:0] is used to define the compensation of D[7:0] and BR_COMP[8] is used to define the parity bit.

Example:

(1). UART's peripheral clock = 32.768K and baud rate is 9600

Baud rate 9600, UART peripheral clock is 32.768K → 3.413 peripheral clock/bit

If the baud divider is set 2 (3 peripheral clock/bit), the inaccuracy of each bit is -0.413 peripheral clock and BR_COMP_DEC =0, so that the BR_COMP (UART_BR_COMP[8:0]) can be set as 9'b001010101 = 0x55

Bit	Name	Total INACCURACY	BR_COMP Compensated	Final Inaccuracy
0	Start	-0.413	x	-0.413
1	D[0]	-0.826(-0.413-0.413)	1	0.176
2	D[1]	-0.237(0.176-0.413)	0	-0.237
3	D[2]	-0.650(-0.237-0.413)	1	0.150
4	D[3]	-0.263(0.150-0.413)	0	-0.263
5	D[4]	-0.676(-0.263-0.414)	1	0.324
6	D[5]	-0.089(0.324-0.413)	0	-0.089
7	D[6]	-0.502(-0.089-0.413)	1	0.498
8	D[7]	0.085(0.498-0.413)	0	0.085
9	Parity	-0.328(0.085-0.413)	0	-0.328

(2). UART's peripheral clock = 32.768K and baud rate is 4800

If the baud divider is set 6 (7 peripheral clock/bit), the inaccuracy of each bit is 0.173 peripheral clock and BR_COMP_DEC =1, so that the BR_COMP (UART_BR_COMP[8:0]) can be set as 9'b010000010 = 0x82.

Bit	Name	Total INACCURACY	BR_COMP Compensated	Final Inaccuracy
0	Start	0.173	x	0.173
1	D[0]	0.346(0.173+0.173)	0	0.346
2	D[1]	0.519(0.346+0.173)	1	-0.481
3	D[2]	-0.308(-0.481+0.173)	0	-0.308
4	D[3]	-0.135(-0.308+0.173)	0	-0.135
5	D[4]	-0.038(-0.135+0.173)	0	0.038
6	D[5]	0.211(0.038+0.173)	0	0.211
7	D[6]	0.384(0.211+0.173)	0	0.384
8	D[7]	0.557(0.384+0.173)	1	-0.443
9	Parity	-0.270(-0.443+0.173)	0	-0.270

6.14.5.4 Auto-Baud Rate Detection

The UART supports auto-baud rate detection. The auto-baud rate detection controls the process of measuring the incoming clock/data rate for the baud rate generation and can be read and written at user discretion. If auto-baud feature enabled, controller will measure the bit time of the received data stream (LSB must be "1") and set the divisor latch registers UART_BARD. Auto-baud rate detection is started by setting the ABAUD_EN (UART_CTL [12]). When the auto-baud rate detection flow finishes, the ABAUD_EN bit will be cleared automatically, and the ABAUD_IS (UART_ISR[7]) and ABAUD_F (UART_TRSR[1]) will be setting. If have time-out occurs (baud rate counter overflow), the ABAUD_IS (UART_ISR[7]) and ABAUD_TOUT_F (UART_TRSR[2]) will be setting. The following diagram demonstrates the auto-baud rate detection function.

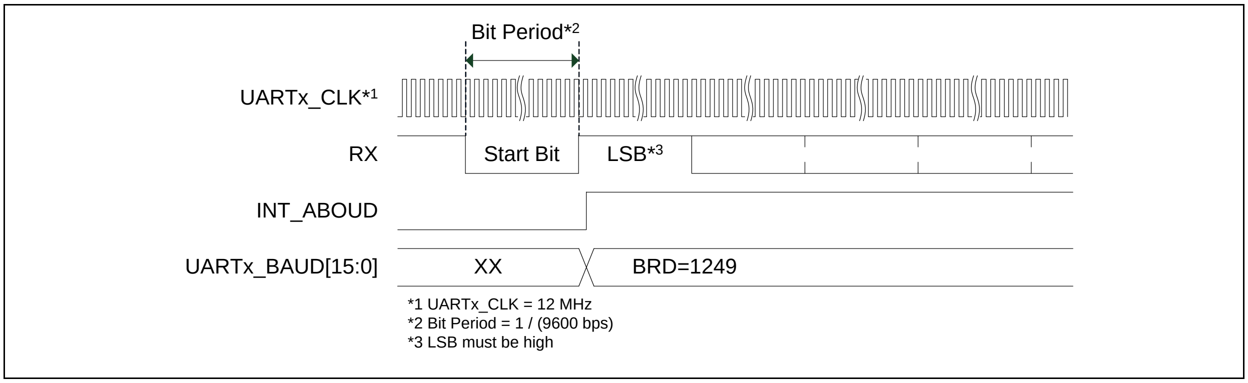


Figure 6‑54 UART Auto-Baud Rate Block Diagram

6.14.5.5 FIFO Control and Status

The UART controller is built-in with a 16-bytes transmitter FIFO (TX_FIFO) and a 16-bytes receiver FIFO (RX_FIFO) that reduces the number of interrupts presented to the CPU. The CPU can read the status of the UART at any time during operation. The reported status information includes the 6 types of interrupts and condition of the transfer operations being performed by the UART, as well as 3 error conditions (parity error, framing error, break interrupt) probably occur while receiving data. This FIFO control and status also support all of UART, IrDA, and RS-485 function mode.

6.14.5.6 Auto-Flow Control

The UART controllers support auto-flow control function that uses two low-level signals, CTSn (clear-to-send) and RTSn (request-to-send) to control the flow of data transfer between the UART and external devices (ex: Modem). When auto-flow is enabled, the UART is not allowed to receive data until the UART asserts RTSn (RTSn high) to external device. When the number of bytes in the RX-FIFO equals the value of RTS_TRI_LEV (UART_TLCTL[13:12]), the RTSn is de-asserted. The UART sends data out when UART controller detects CTSn is asserted (CTSn high) from external device. If a valid asserted CTSn is not detected the UART controller will not send data out.

The following diagram demonstrates the auto-flow control block diagram.

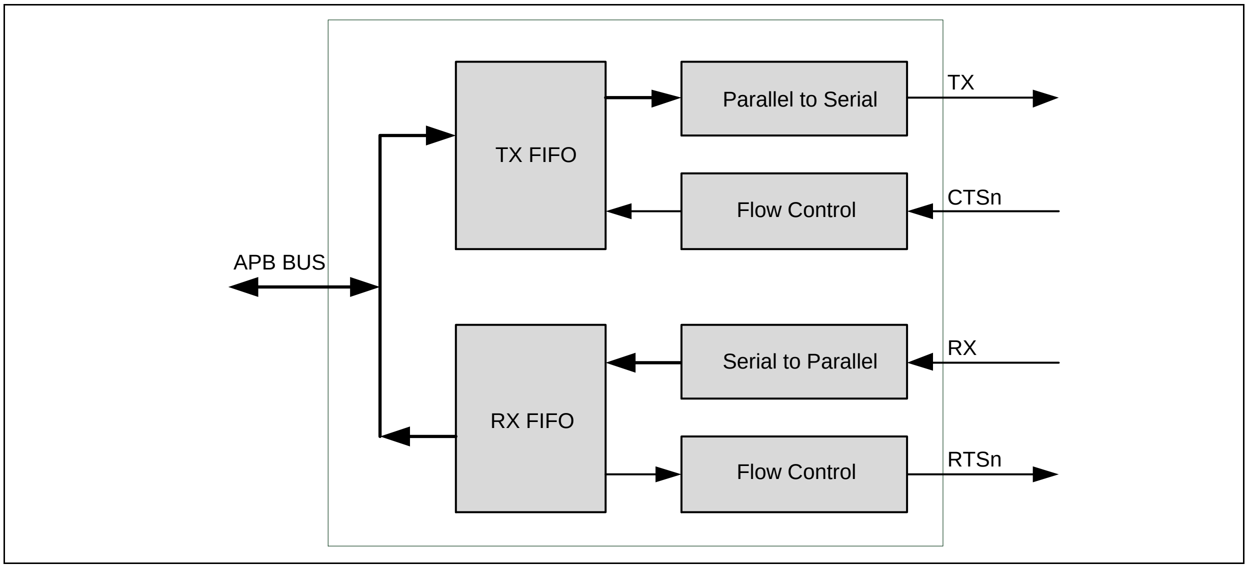


Figure 6‑55 UART Auto-Flow Control Block Diagram

6.14.5.7 Wake-Up Function

The UART Controller supports wake-up system function. The wake-up source includes:

(a) CTSn pin wake-up (WAKE_CTS_EN (UART_CTL[8]))

(b) Incoming data wake-up (WAKE_DATA_EN (UART_CTL[9]))

(c) Received Data FIFO reached threshold wake-up (WAKE_THRESH_EN (UART_CTL[17]))

(d) RS-485 Address Match (AAD mode) wake-up(WAKE_RS-485 AAD_EN (UART_CTL[18]))

Each wake-up source description as following:

(a) CTSn pin wake-up

When system is in power-down and the WAKE_CTS_EN (UART_CTL[8]) is set , the toggle of CTSn pin can wake-up the system.

Case 1:

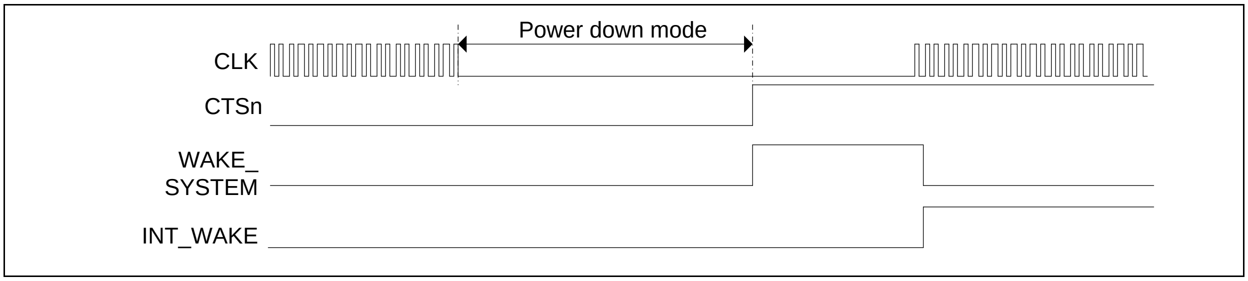


Figure 6‑56 UART CTSn Wake-Up Case 1

Case 2:

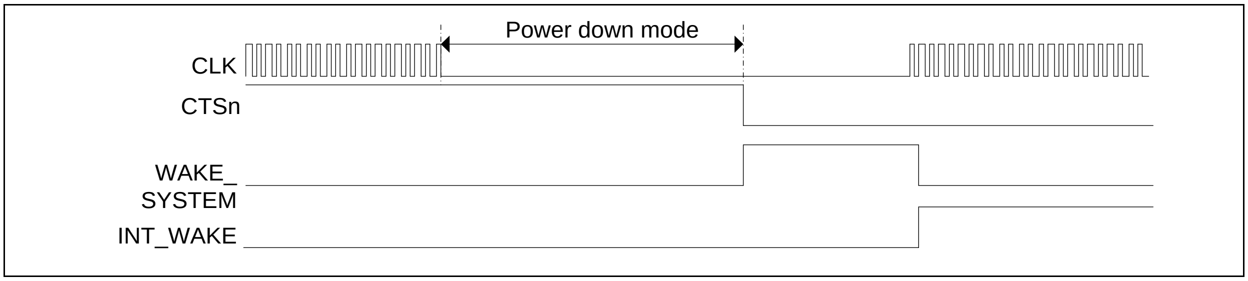


Figure 6‑57 UART CTSn Wake-Up Case 2

(b) Incoming data wake-Up

When system is in power-down and the WAKE_DATA_EN (UART_CTL[9]) is set, the toggle of incoming data (SIN pin) pin can wake-up the system.

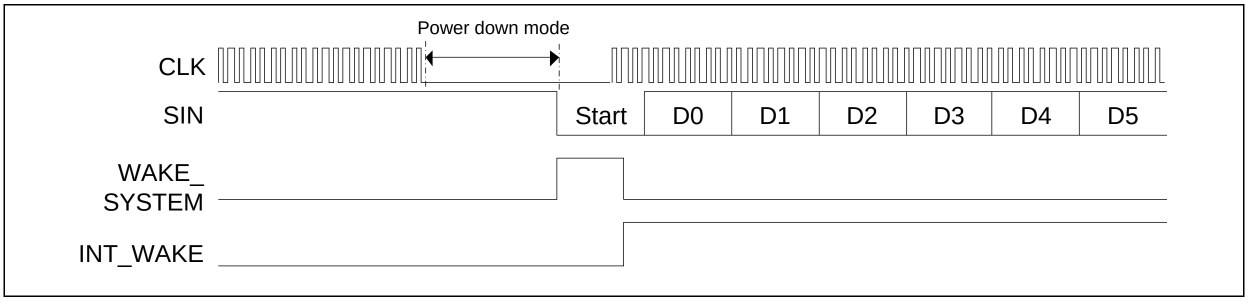


Figure 6‑58 UART DATA Wake-UI

(c). Received Data FIFO reached threshold wake-up

In power down condition, when the FIFO threshold reached wake-up function enable bit WAKE_THRESH_EN (UART_CTL[17]) is set and the received data FIFO reached the threshold value RFITL (UART_TLCTL[9:8]) can wake-up the system. When system wakes up, controller will clear the WAKE_DATA_EN (UART_CTL[9]) register automatically.

Note: Both of the Clock Divider of APBDIV (APB_DIV[2:0]) and the HCLK_N (CLK_DIV0[3:0]) shall be set as 0 in this condition.

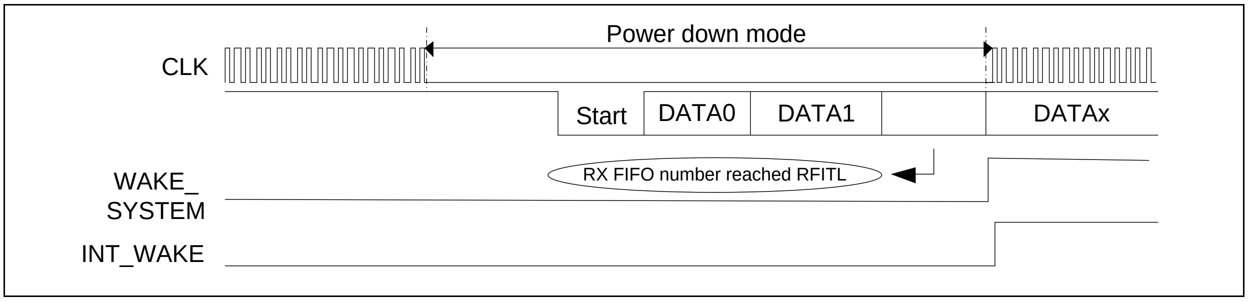


Figure 6‑59 UART Receive DATA FIFO Reach RFITL Wake-Up

(d) RS-485 Address Match (AAD mode) wake-up

In power down condition, when the WAKE_RS485_AAD_EN (UART_CTL[18]) is set and the received data match the ADDR_PID_MATCH (UART_ALT_CTL[28:24]), it can wake-up the system.

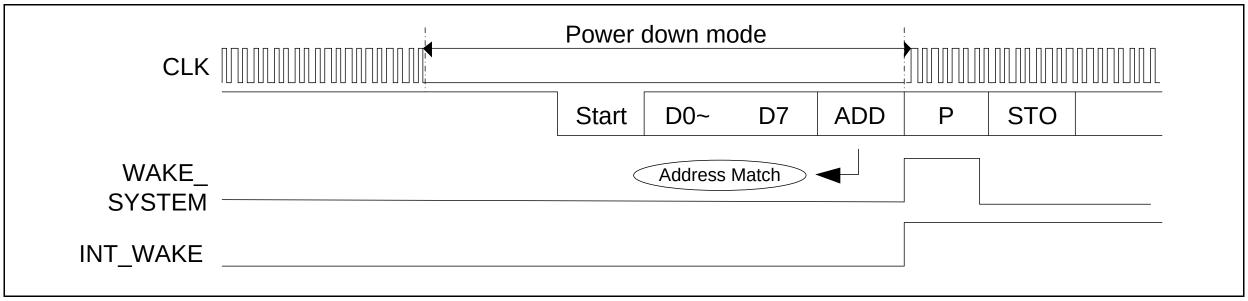


Figure 6‑60 UART RS-485 AAD Mode Address Match Wake-Up

6.14.5.8 IrDA Function Mode

The UART Controller provides Serial IrDA (SIR, Serial Infrared) transmit encoder and receive decoder function. The FUN_SEL (UART_FUN_SEL[1:0]) bits are used to select IrDA function.

The SIR specification defines a short-range infrared asynchronous serial transmission mode with one start bit, 8 data bits, and 1 stop bit. The maximum data rate is 115.2 Kbps (half duplex). The IrDA SIR block contains an IrDA SIR Protocol encoder/decoder. The IrDA SIR protocol is half-duplex only. So it cannot transmit and receive data at the same time. The IrDA SIR physical layer specifies a minimum 10 ms transfer delay between transmission and reception.

In IrDA Operation mode, the receive FIFO trigger level must be “1” by setting RFITL (UART_TLCTL[9:8]) = “0” and the DIV_16_EN (UART_BAUD[31]) bit must be enabled in IrDA mode operation (Mode 1).

Baud Rate = Clock / (16 * (BRD + 1)), where BRD is Baud Rate Divider in UART_BAUD register.

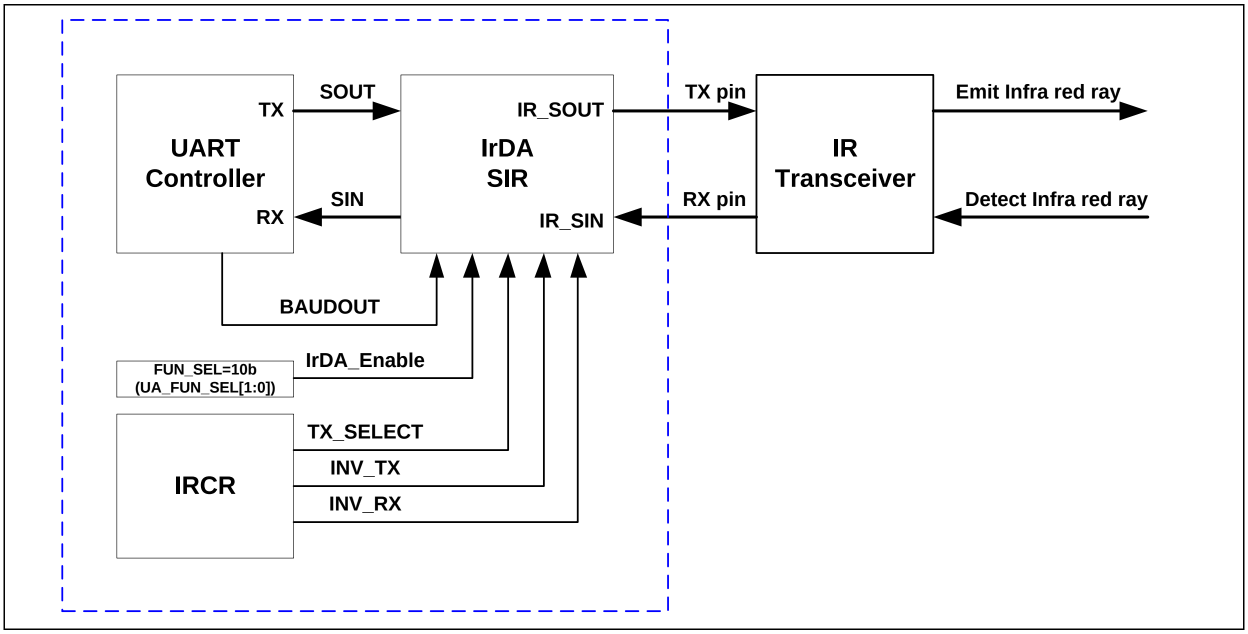


Figure 6‑61 IrDA Block Diagram

IrDA SIR Transmit Encoder

The IrDA SIR Transmit Encoder modulates Non-Return-to Zero (NRZ) transmit bit stream output from UART. The IrDA SIR physical layer specifies the usage of Return-to-Zero, Inverted (RZI) modulation scheme which represent logic 0 as an infra light pulse. The modulated output pulse stream is transmitted to an external output driver and infrared Light Emitting Diode.

The transmitted pulse width is specified as 3/16 period of baud rate.

IrDA SIR Receive Decoder

The IrDA SIR Receive Decoder demodulates the return-to-zero bit stream from the input detector and outputs the NRZ serial bits stream to the UART received data input. The decoder input is normally high in the idle state. (Because of this, IRCR bit 6 should be set as “1” by default)

A start bit is detected when the decoder input is LOW

IrDA SIR Operation

The IrDA SIR Encoder/decoder provides functionality which converts between UART data stream and half duplex serial SIR interface. Refer to UART_IRCR register for detail description. The following diagram is IrDA encoder/decoder waveform:

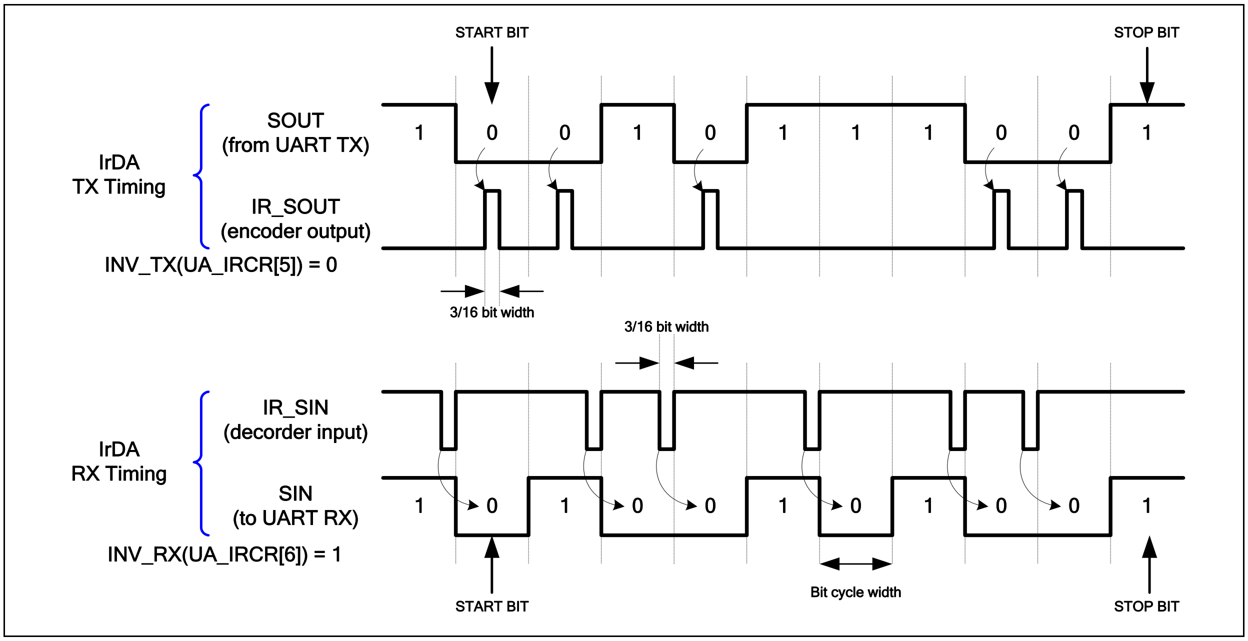


Figure 6-62 IrDA TX/RX Timing Diagram

6.14.5.9 RS-485 Function Mode

Another alternate function of UART controllers is RS-485 9 bit mode function whose direction control can be controlled by RTSn pin or GPIO. The RS-485 mode is selected by setting the UART_FUN_SEL register to select RS-485 function and when operating in RS-485 mode, the receive FIFO trigger level must be “1” by setting RFITL (UART_TLCTL[9:8]) = “0”. The RS-485 driver control is implemented by the RTSn control signal from an asynchronous serial port to enable the RS-485 driver. In RS-485 mode, many characteristics of the RX and TX are same as UART.

In RS-485 function mode, the bit 9 will be configured as address bit. For data characters, the bit 9 is set to “0”. Software can program UART_TLCTL register to control the 9-th bit (When the PBE, EPE and SPE are set, the 9-th bit is transmitted as “0” and when PBE and SPE are set and EPE is cleared, the 9-th bit is transmitted as “1”). The Controller supports three operation modes that are RS-485 Normal Multi-drop Operation Mode (RS-485 NMM Mode), RS-485 Auto Address Detection Operation Mode

(RS-485 AAD Mode) and RS-485 Auto Direction Control Operation Mode (RS-485 AUD Mode). One of the three operation modes can be selected by programming UART_ALT_CTL register, and software can driving the transfer delay time between the last stop bit leaving the TX-FIFO and the de-assertion by setting DLY (UART_TMCTL[23:16]) register.

RS-485 Normal Multi-drop Operation Mode (RS-485 NMM Mode)

In RS-485 Normal Multi-drop operation mode, software can decide whether receiver will ignore data before an address byte is detected (bit 9 = “1”). When an address byte be detected (bit 9 = “1”) by hardware, the address byte data will be stored in the RX-FIFO. Software can decide whether to enable or disable receiver to accept the following data byte by setting UART_CTL [RX_DIS]. If the receiver is be enabled (RX_DIS (UART_CTL[2]) is low), all received byte data will be accepted and stored in the RX-FIFO, and if the receiver is disabled(RX_DIS (UART_CTL[2]) is high), all received byte data will be ignore until the next address byte be detected. If software disable receiver by setting RX_DIS (UART_CTL[2]) register high, when a next address byte is detected, the controller will clear the RX_DIS (UART_CTL[2]) bit and the address byte data will be stored in the RX-FIFO.

Program Sequence Example:

1. Program FUN_SEL (UART_FUN_SEL[1:0]) to select RS-485 function.
2. Program the RX_DIS bit (UART_CTL[2]) to determine whether to store the received data before an address byte is detected (bit 9 = “1”).
3. Program the RS-485_NMM by setting UART_ALT_CTL register.
4. When an address byte is detected (bit 9 = “1”), hardware will set RLS_IS (UART_ISR[2]) and RS-485_ADDET_F (UART_TRSR[0]) flag.
5. Software can decide whether to accept the following data byte by setting RX_DIS (UART_CTL[2]).
6. Repeat step 4 and step 5.

RS-485 Auto Address Detection Operation Mode (RS-485 AAD Mode)

In RS-485 Auto Address Detection Operation Mode, the receiver will ignore any data until an address byte is detected (bit 9 = “1”) and the address byte data match the ADDR_PID_MATCH (UART_ALT_CTL[31:24]) value. The address byte data will be stored in the RX-FIFO. The following all data will be accepted and stored in the RX-FIFO until an address byte not match the ADDR_PID_MATCH (UART_ALT_CTL[31:24]) value. In RS-485 AAD mode, don’t fill any value to RX_DIS (UART_CTL[2]) bit.

Program Sequence example:

1. Program FUN_SEL (UART_FUN_SEL[1:0]) to select RS-485 function.
2. Program the RS-485_AAD (UART_ALT_CTL[17]).
3. When an address byte is detected (bit9 = “1”), hardware will compare the address byte and the ADDR_PID_MATCH (UART_ALT_CTL[31:24]) value.
4. If the address byte matches the ADDR_PID_MATCH (UART_ALT_CTL[31:24]) value, hardware will set RLS_IS (UART_ISR[2]) and RS-485_ADDET_F flag (UART_TRSR[0]). And the receiver will sorted address byte to FIFO and accept the following data transfer and stored data in FIFO until next address byte be detected.

   However if the address byte does not match the ADDR_PID_MATCH (UART_ALT_CTL[31:24]) value, hardware will ignored the address byte data and ignored the following data transfer.
5. Respect step 3 and step 4.

RS-485 Auto Direction Mode (RS-485 AUD Mode)

Another option function of RS-485 controllers is RS-485 auto direction control function. The RS-485 driver control is implemented by using the RTSn control signal from an asynchronous serial port to enable the RS-485 driver. The RTSn line is connected to the RS-485 driver enable such that setting the RTSn line to high (logic “1”) will enable the RS-485 driver. Setting the RTSn line to low (logic “0”) will put the driver into the tri-state condition. User can setting LEV_RTS in UART_MCSR register to change the RTSn driving level.

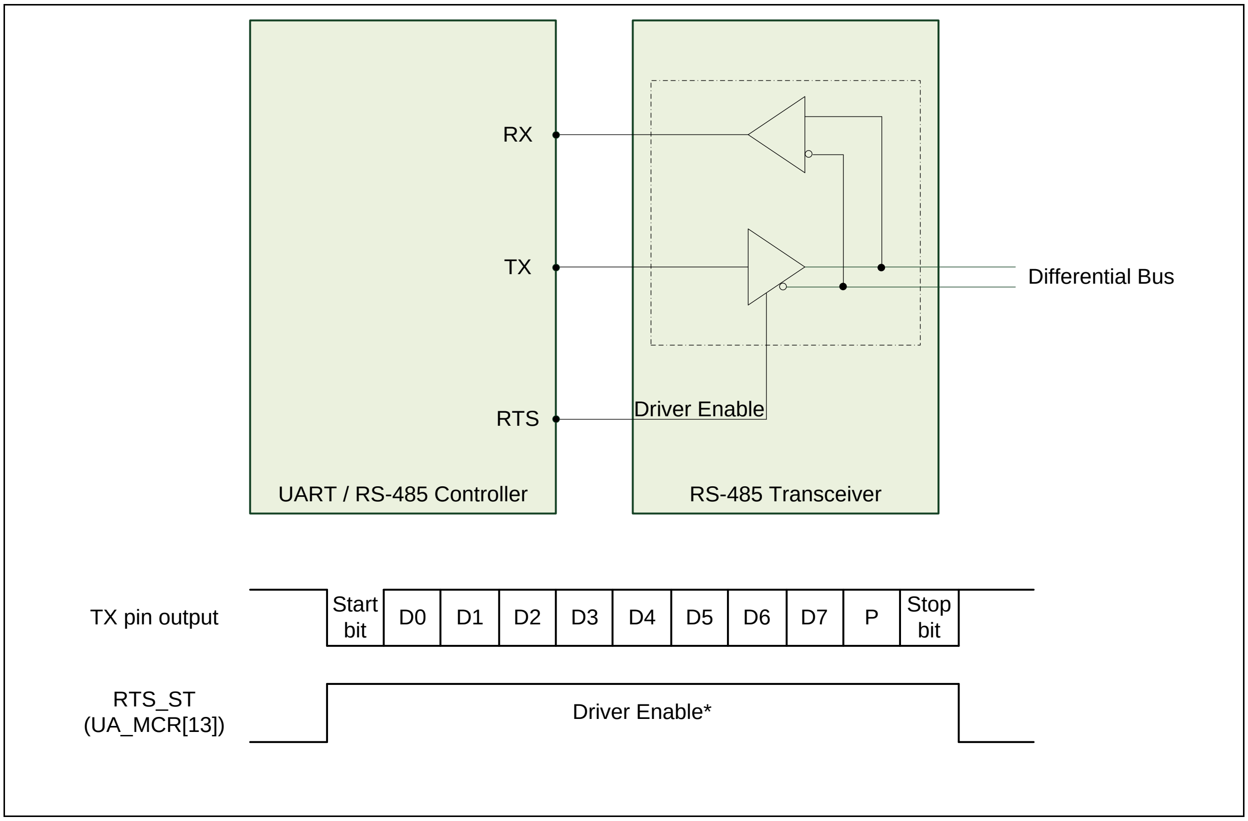


Figure 6-63 Structure of RS-485 Frame

6.14.5.10 LIN (Local Interconnection Network) Function Mode

The UART Controller supports LIN function mode. The LIN mode is selected by setting the (UART_FUN_SEL[1:0] = 1). In LIN function mode, each byte field is initialed by a start bit with value 0 (dominant), followed by 8 data bits (LSB is first) and ended by 1 stop bit with value one (recessive) in accordance with the LIN standard.

There is LIN received interrupt when the LIN header field is received. The header includes (1). “break field” (2). “break field + sync field” and (3). “break field + sync field + PID field”. All of them can be choose by setting UART_ALT_CTL [LIN_HEAD_SEL] register.

(1). If the field includes “break field”, when the receiver received break field then the LIN_RX_F will be set. The controller will receive next data and put it in FIFO.

(2). If the field includes “break field + sync field”, hardware will wait for the flag LIN_RX_F in UART_TRSR to check RX received break field and sync field. If the break and sync field is received, hardware will set UART_TRSR [LIN_RX_F] flag, and if the break is received but the sync field does not equal 0x55, then hardware will set UART_TRSR [LIN_RX_F] and UART_TRSR

[LIN_RX_SYNC_ERR_F] flag. The break and sync data (equals 0x55 or not) will not be stored in FIFO.

(3). If the field includes "break field + sync field + PID field", In this operation mode, hardware will control data automatically. Hardware will ignore any data until received break + sync (0x55) + PID value match the UART_ALT_CTL [ADDR_MATCH] value (break + sync + PID will not be stored in FIFO).

When received break + sync (0x55) + PID value match the UART_ALT_CTL [ADDR_MATCH] value, hardware will set UART_TRSR [LIN_RX_F] and the following all data will be accepted and stored in the RX-FIFO until detect next break field.

If the receiver received break + wrong sync (not equal 0x55) + PID value, hardware will set UART_TRSR [LIN_RX_F] and UART_TRSR [LIN_RX_SYNC_ERR_F] flag and the receiver will be disabled. If the receiver received break + sync (0x55) + wrong PID value, hardware will set UART_TRSR [LIN_RX_F] flag and the receiver will be disabled.

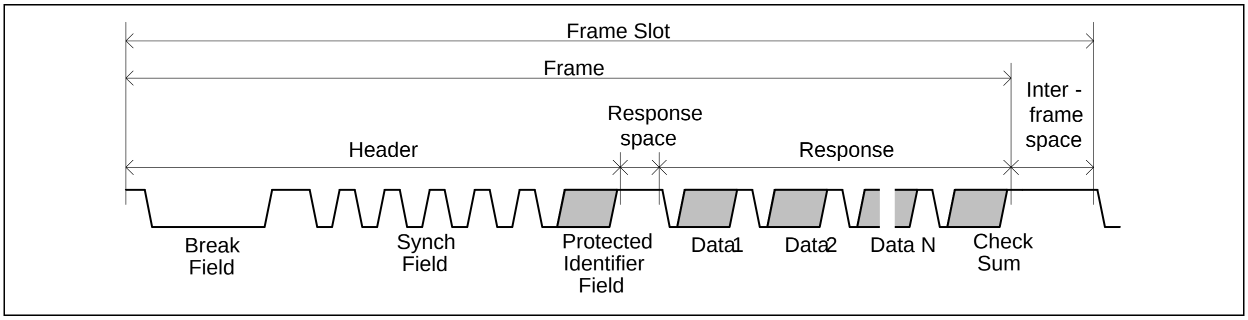


Figure 6-64 Structure of LIN Frame

Program Flow of LIN Bus Transmit Transfer (TX)

The program flow of LIN Bus Transmit transfer (TX) is shown as follows

Case 1: The Header Field Select as "Break"

1. Set LIN function mode by setting UART_FUN_SEL register.
2. Choose the data header to "break field" by setting LIN_HEAD_SEL (UART_ALT_CTL[5:4]).
3. Enable BIT_ERR_EN bit (UART_ALT_CTL[8]), and when the SIN pin is not equal to SOUT pin that the hardware will generator an interrupt to CPU). If user wants to receive data at the same time, user must enable LIN_RX_EN bit (UART_ALT_CTL[6]).
4. Fill LIN_TX_BCNT to choose break field length. (The break field length is LIN_TX_BCNT + 8).
5. Set the LIN_TX_EN bit (UART_ALT_CTL[7]) register to start transmission break field, and when break filed operation is finished, hardware will set LIN_IS (UART_ISR[8]) and LIN_TX_F (UART_TRSR[3]) flag and LIN_TX_EN (UART_ALT_CTL[7]) will be cleared automatically.
6. Fill 0x55 to UART_THR to request synch field transmission.
7. Fill the protected identifier value (PID) in the UART_THR
8. Fill N bytes data and Checksum to UART_THR then repeat step 4 ~ step 8 to transmit the data.

Case 2: The Header Field Select as "Break + Sync"

1. Set LIN function mode by setting UART_FUN_SEL register.

2. Choose the data header to “break + sync” field by setting LIN_HEAD_SEL (UART_ALT_CTL[5:4]).
3. Enable BIT_ERR_EN bit (UART_ALT_CTL[8]), and when the SIN pin is not equal to SOUT pin that the hardware will generator an interrupt to CPU). If user wants to receive data at the same time, user must enable LIN_RX_EN bit (UART_ALT_CTL[6]).
4. Fill LIN_TX_BCNT (UART_ALT_CTL[2:0]) to choose break field length. (The break field length is LIN_TX_BCNT + 8).
5. Set the LIN_TX_EN bit (UART_ALT_CTL[7]) to start transmission break and sync field, and when break and sync filed operation is finished, hardware will set LIN_IS (UART_ISR[2]) and LIN_TX_F flag (UART_TRSR[3]) and LIN_TX_EN (UART_ALT_CTL[7]) will be cleared automatically.
6. Fill the protected identifier value (PID) in the UART_THR
7. Fill N bytes data and Checksum to UART_THR then repeat step 4 ~ step 7 to transmit the data.
8.

Case 3: The Header Field Select as “Break + Sync + PID”

1. Set LIN function mode by setting UART_FUN_SEL register.
2. Choose the data header to “break + sync + PID field” by setting LIN_HEAD_SEL (UART_ALT_CTL[5:4]).
3. Enable BIT_ERR_EN bit (UART_ALT_CTL[8]), and when the SIN pin is not equal to SOUT pin that the hardware will generator an interrupt to CPU). If user wants to receive data at the same time, user must enable LIN_RX_EN bit (UART_ALT_CTL[6]).
4. Fill LIN_TX_BCNT (UART_ALT_CTL[7]) to choose break field length. (The break field length is LIN_TX_BCNT + 8).
5. Set the LIN_TX_EN bit (UART_ALT_CTL[7]) to start transmission break, sync and PID field, and when break, sync and PID filed operation is finished, hardware will set LIN_IS (UART_ISR[2]) and LIN_TX_F flag (UART_TRSR[3]) and LIN_TX_EN (UART_ALT_CTL[7]) will be cleared automatically.
6. Fill N bytes data and Checksum to UART_THR then repeat step 4 ~ step 6 to transmit the data.

Program Flow of LIN Bus Receiver transfer (RX)

The program flow of LIN Bus Receiver transfer (RX) is show as follows.

Case 1: The header Field Select as “Break”

1. Set LIN function mode by setting UART_FUN_SEL register.
2. Choose the data header to “break field” by setting LIN_HEAD_SEL (UART_ALT_CTL[5:4]).
3. Set the LIN_RX_EN bit (UART_ALT_CTL[6]) to enable LIN RX mode.
4. Wait for the flag LIN_RX_F (UART_TRSR[4]) to check RX received break field or not. (The break field will not be stored in FIFO)
5. Wait for the flag RDA_IF (UART_ISR[0]) and read back the UART_RBR register.

Case 2: The Header Field Select as “Break + Sync”

1. Set LIN function mode by setting UART_FUN_SEL register.

2. Choose the data header to “break + sync field” by setting LIN_HEAD_SEL (UART_ALT_CTL [5:4]).
3. Set the LIN_RX_EN bit (UART_ALT_CTL[6]) to enable LIN RX mode.
4. Wait for the flag LIN_RX_F in UART_TRSR to check RX received break field and sync field. If the break and sync field is received, hardware will set LIN_RX_F (UART_TRSR[4]) flag If the break be received but the sync field not equal 0x55, hardware will set LIN_RX_F (UART_TRSR[4]) and LIN_RX_SYNC_ERR_F flag (UART_TRSR[8]). The break and sync data (equal 0x55 or not) will not be stored in FIFO.
5. Wait for the flag RDA_IF (UART_ISR[0]) and read back the UART_RBR register.

Case 3: The Header Field Select as “Break + Sync + PID”

1. Set LIN function mode by setting UART_FUN_SEL register.
2. Choose the data header to “break + sync + PID field” by setting LIN_HEAD_SEL (UART_ALT_CTL[5:4]).
3. Set the LIN_RX_EN bit (UART_ALT_CTL[6]) to enable LIN RX mode.
4. In this operation mode, hardware will control data automatically. Hardware will ignore any data until received break + sync (0x55) + PID value match the ADDR_PID_MATCH (UART_ALT_CTL[31:24]) value (break + sync + PID will not be stored in FIFO). When received break + sync (0x55) + PID value match the ADDR_PID_MATCH (UART_ALT_CTL[31:24]) value, hardware will set LIN_RX_F (UART_TRSR[4]) and the following all data will be accepted and stored in the RX-FIFO until detect next break field. If the receiver received break + wrong sync (not equal 0x55) + PID value, that hardware will set LIN_RX_F flag (UART_TRSR[4]) and LIN_RX_SYNC_ERR_F flag (UART_TRSR[8]) and the receiver will be disabled. If the receiver received break + sync (0x55) + wrong PID value, that hardware will set LIN_RX_F flag (UART_TRSR[4]) and the receiver will be disabled.
5. Wait for the flag RDA_IF (UART_ISR[0]) and read back the UART_RBR register.

6.14.5.11 Interrupt

The UART Controller supports nine types of interrupts including

(1). Receiver threshold level reaching interrupt (INT_RDA)

(2). Transmitter FIFO empty interrupt (INT_THRE)

(3). Line status interrupt (break error, parity error, framing error or RS-485 interrupt) (INT_RLS)

(4). time-out interrupt (INT_TOUT)

(5). MODEM status interrupt (INT_MODEM)

(6). Buffer error interrupt (INT_BUF_ERR)

(7). wake-up interrupt (INT_WAKE)

(8). auto-baud rate detect or auto-baud rate counter overflow flag (INT_ABAUD)

(9). LIN function interrupt (INT_LIN)

Interrupt Indicator	Interrupt Source	Interrupt Enable	Interrupt Flag	Flag Clear
INT_LIN	LIN Function Interrupt	BUF_ERR_IE	LIN_IS(UART_ISR[8]) = BIR_ERR_F(UART_TRSR[5]) or LIN_RX_F(UART_TRSR[4]) or	Write “1” to BIR_ERR_F(UART_TRSR [5]) or LIN_RX_F(UART_TRSR [4]) or

			LIN_TX_F(UART_TRSR [3])	LIN_TX_F(UART_TRSR [3]).
INT_ABAUD	Auto-Baud Rate Interrupt	ABAUD_IE	ABAUD_IS(UART_ISR[7]) = ABAUD_TOUT_F(UART_TRSR [2]) or ABAUD_F(UART_TRSR [1])	Write “1” to ABAUD_TOUT_F(UART_TRSR [7]), ABAUD_F(UART_TRSR[1]).
INT_WAKE	Wake-Up Interrupt	WAKE_IE	WAKE_IS(UART_ISR[6])	Write “1” to WAKE_IS(UART_ISR[6])
INT_BUF_ERR	Buffer Error Interrupt	BUF_ERR_IE	BUF_ERR_IS(UART_ISR[5]) = RX_OVER_F(UART_FSR [0]) or TX_OVER_F(UART_FSR [8]).	Write “1” to RX_OVER_F(UART_FSR [0]) or TX_OVER_F(UART_FSR [8])
INT_RTO	RX Time-Out Interrupt	RTO_IE	RTO_IS(UART_)SR[4]]	Read UART_RBR
INT_MODEM	Modem Status Interrupt	MODEM_IE	MODME_IS(UART_ISR[3]) = DCT_F(UART_MCSR[18])	Write “1” to DCT_F(UART_MCSR[18])
INT_RLS	Receive Line Status Interrupt	RLS_IE	RLS_IS(UART_ISR[2])= BI_F(UART_FSR [6]) or FE_F(UART_FSR [5]) or PE_F(UART_FSR [4]) or RS-485_ADDDET_F(UART_TRSR [0])	Write “1” to BI_F(UART_FSR [6]) or FE_F(UART_FSR [5]) or PE_F(UART_FSR [4]) or RS-485_ADDDET_F(UART_TRSR[0])
INT_THRE	Transmit Holding Register Empty Interrupt	THRE_IE	THRE_IS(UART_ISR[1])	Write UART_THR
INT_RDA	Receive Data Available Interrupt	RDA_IE	RDA_IS(UART_ISR[0])	Read UART_RBR

Table 6-19 UART Interrupt Sources and Flags

6.14.5.12 Time-out

The UART Controller supports 9-bits time-out counter. The time-out counter resets and starts counting (the counting clock = baud rate) whenever the RX-FIFO receives a new data word. Once the content of time-out counter (TOUT_CNT) is equal to time-out interrupt comparator (TOIC (UART_TMCTL[8:0])), a receiver time-out interrupt (INT_TOUT) is generated if UART_IER [RTO_IEN]. A new incoming data word or RX-FIFO empty clears INT_TOUT. Fill all “0” to the TOIC, the time-out function is disabled and the real time-out value is TOIC + 1.

6.14.5.13 PWM modulation

The UART Controller supports UART transmit bus to modulate with PWM channel. The PWM channel selection is defined in PWM_SEL (UART_CTL[26:24]).

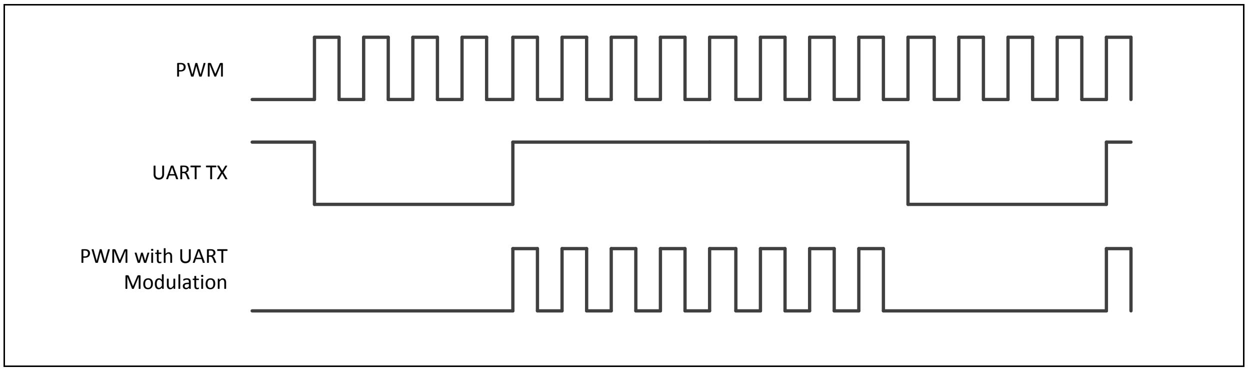


Figure 6-65 UART with PWM Modulation Waveform

6.14.6 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
UART Base Address: **UARTx_BA = 0x4005_0000 + 0x100000*x** **x=0,1**				
UART_RBR	UARTx_BA+0x00	R	UART Receive Buffer Register.	Undefined
UART_THR	UARTx_BA+0x00	W	UART Transmit Holding Register.	Undefined
UART_CTL	UARTx_BA+0x04	R/W	UART Control State Register.	0x0700_0000
UART_TLCTL	UARTx_BA+0x08	R/W	UART Transfer Line Control Register.	0x0000_0000
UART_IER	UARTx_BA+0x0C	R/W	UART Interrupt Enable Register.	0x0000_0000
UART_ISR	UARTx_BA+0x10	R/W	UART Interrupt Status Register.	0x0000_0002
UART_TRSR	UARTx_BA+0x14	R/W	UART Transfer State Status Register.	0x0000_0000
UART_FSR	UARTx_BA+0x18	R/W	UART FIFO State Status Register.	0x0000_0A02
UART_MCSR	UARTx_BA+0x1C	R/W	UART Modem State Status Register.	0x0002_0002
UART_TMCTL	UARTx_BA+0x20	R/W	UART Time-Out Control State Register.	0x0000_01FF
UART_BAUD	UARTx_BA+0x24	R/W	UART Baud Rate Divisor Register	0x0000_0000
UART_IRCR	UARTx_BA+0x30	R/W	UART IrDA Control Register.	0x0000_0040
UART_ALT_CSR	UARTx_BA+0x34	R/W	UART Alternate Control State Register.	0x0000_0000
UART_FUN_SEL	UARTx_BA+0x38	R/W	UART Function Select Register.	0x0000_0000
UART_BR_COMP	UARTx_BA+0x3C	R/W	UART Baud Rate Compensation	0x0000_0000

Note: The x of the UARTx_REG represents the UART channel.

6.14.7 Register Description

UART Receive Buffer Register (UARTx_RBR)

Register	Offset	R/W	Description	Reset Value
UART_RBR	UARTx_BA+0x00	R	UART Receive Buffer Register.	Undefined

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RBR							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**RBR**	**Receive Buffer Register** By reading this register, the UART will return an 8-bit data received from RX pin (LSB first).

UART Transmit Holding Register (UARTx_THR)

Register	Offset	R/W	Description	Reset Value
UART_THR	UARTx_BA+0x00	W	UART Transmit Holding Register.	Undefined

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
THR							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**THR**	**Transmit Holding Register** By writing to this register, the UART will send out an 8-bit data through the TX pin (LSB first).

UART Control Register (UARTx_CTL)

Register	Offset	R/W	Description	Reset Value
UART_CTL	UARTx_BA+0x04	R/W	UART Control State Register.	0x0700_0000

31	30	29	28	27	26	25	24
Reserved					PWM_SEL		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved					WAKE_RS485_AAD_EN	WAKE_THRESH_EN	Reserved
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			ABAUD_EN	Reserved		WAKE_DATA_EN	WAKE_CTS_EN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMA_TX_EN	DMA_RX_EN	AUTO_CTS_EN	AUTO_RTS_EN	TX_DIS	RX_DIS	TX_RST	RX_RST

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26:24]	**PWM_SEL**	**PWM Channel Select for Modulation** Select the PWM channel to modulate with the UART transmit bus. 000 = PWM channel 0 modulate with UART TX. 001 = PWM channel 1 modulate with UART TX. 010 = PWM channel 2 modulate with UART TX. 011 = PWM channel 3 modulate with UART TX. The others, no modulation of UART with PWM
[23:19]	**Reserved**	Reserved.
[18]	**WAKE_RS485_AAD_EN**	**RS-485 Address Match Wake-up Function Enable Control** 0 = RS-485 ADD mode address match wake-up function Disabled. 1 = RS-485 AAD mode address match wake-up function Enabled when the system is in Power-down mode.
[17]	**WAKE_THRESH_EN**	**FIFO Threshold Reach Wake-up Function Enable Control** 0 = Received FIFO threshold reach wake-up function Disabled. 1 = Received FIFO threshold reach wake-up function Enabled when the system is in Power-down mode.
[16:13]	**Reserved**	Reserved.
[12]	**ABAUD_EN**	**Auto-baud Rate Detect Enable Control** 0 = Auto-baud rate detect function Disabled. 1 = Auto-baud rate detect function Enabled. **Note:** When the auto-baud rate detect operation finishes, hardware will clear this bit and the associated interrupt (INT_ABAUD) will be generated (If ABAUD_IE (UART_IER [7]) be enabled).
[11:10]	**Reserved**	Reserved.

Bits	Description	
[9]	**WAKE_DATA_EN**	**Incoming Data Wake-up Function Enable Control** 0 = Incoming data wake-up function Disabled. 1 = Incoming data wake-up function Enabled when the system is in Power-down mode, incoming data will wake-up system from Power-down mode. **Note:** Hardware will clear this bit when the incoming data wake-up operation finishes and “system clock” work stable.
[8]	**WAKE_CTS_EN**	**CTSn Wake-up Function Enable Control** When the system is in Power-down mode, an external CTSn change will wake-up system from Power-down mode. 0 = CTSn wake-up function Disabled. 1 = CTSn wake-up function Enabled.
[7]	**DMA_TX_EN**	**TX DMA Enable Control** 0 = TX PDMA service function Disabled. 1 = TX PDMA service function Enabled.
[6]	**DMA_RX_EN**	**RX DMA Enable Control** 0 = RX PDMA service function Disabled. 1 = RX PDMA service function Enabled.
[5]	**AUTO_CTS_EN**	**CTSn Auto-flow Control Enable Control** When CTSn auto-flow is enabled, the UART will send data to external device when CTSn input assert (UART will not send data to device until CTSn is asserted). 0 = CTSn auto-flow control Disabled. 1 = CTSn auto-flow control Enabled.
[4]	**AUTO_RTS_EN**	**RTSn Auto-flow Control Enable Control** When RTSn auto-flow is enabled, if the number of bytes in the RX-FIFO equals the RTS_TRI_LEV (UART_TLCTL[13:12]), the UART will reassert RTSn signal. 0 = RTSn auto-flow control. Disabled. 1 = RTSn auto-flow control Enabled.
[3]	**TX_DIS**	**Transfer Disable Control** 0 = Transfer Enabled. 1 = Transfer Disabled.
[2]	**RX_DIS**	**Receiver Disable Control** 0 = Receiver Enabled. 1 = Receiver Disabled. **Note1:** In RS-485 NMM mode, user can set this bit to receive data before detecting address byte. **Note2:** In RS-485 AAD mode, this bit will be setting to “1” automatically. **Note3:** In RS-485 AUD mode and LIN “break + sync +PID” header mode, hardware will control data automatically, so don’t fill any value to this bit.
[1]	**TX_RST**	**TX Software Reset** When TX_RST is set, all the bytes in the transmitting FIFO and TX internal state machine are cleared. 0 = No effect. 1 = Reset the TX internal state machine and pointers. **Note:** This bit will be auto cleared and take at least 3 UART engine clock cycles.
[0]	**RX_RST**	**RX Software Reset** When RX_RST is set, all the bytes in the receiving FIFO and RX internal state machine are cleared.

Bits	Description	
		0 = No effect. 1 = Reset the RX internal state machine and pointers. **Note:** This bit will be auto cleared and take at least 3 UART engine clock cycles.

UART Transfer Line Control Register (UARTx_TLCTL)

Register	Offset	R/W	Description	Reset Value
UART_TLCTL	UARTx_BA+0x08	R/W	UART Transfer Line Control Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**RTS_TRI_LEV**		**Reserved**		**RFITL**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**BCB**	**SPE**	**EPE**	**PBE**	**NSB**	**DATA_LEN**	

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13:12]	**RTS_TRI_LEV**	**RTSn Trigger Level (For Auto-flow Control Use)** 00 = RTS Trigger Level is 1 byte. 01 = RTS Trigger Level is 4 bytes. 10 = RTS Trigger Level is 8 bytes. 11 = RTS Trigger Level is 14 bytes. **Note:** This field is used for auto RTSn flow control.
[11:10]	**Reserved**	Reserved.
[9:8]	**RFITL**	**RX-fIFO Interrupt (INT_RDA) Trigger Level** When the number of bytes in the receiving FIFO is equal to the RFITL then the RDA_IF will be set (if RDA_IE(IER[0]) is enabled, an interrupt will be generated) 00 = RX FIFO Interrupt Trigger Level is 1 byte. 01 = RX FIFO Interrupt Trigger Level is 4 bytes. 10 = RX FIFO Interrupt Trigger Level is 8 bytes. 11 = RX FIFO Interrupt Trigger Level is 14 bytes. **Note:** When operating in IrDA mode or RS-485 mode, the RFITL must be set to “0”.
[7]	**Reserved**	Reserved.
[6]	**BCB**	**Break Control Bit** When this bit is set to logic “1”, the serial data output (TX) is forced to the Spacing State (logic “0”). This bit acts only on TX pin and has no effect on the transmitter logic. 0 = Break control Disabled. 1 = Break control Enabled.
[5]	**SPE**	**Stick Parity Enable Control** 0 = Stick parity Disabled. 1 = Stick parity Enabled. **Note1:** When bits PBE, EPE and SPE are set, the parity bit is transmitted and checked as “0”. When PBE and SPE are set and EPE is cleared, the parity bit is transmitted and

Bits	Description	
		checked as “1”. **Note2:** In RS-485 mode, PBE, EPE and SPE can control bit 9, the bit 9 setting are shown as follows.
[4]	**EPE**	**Even Parity Enable Control** 0 = Odd number of logic 1’s are transmitted or check the data word and parity bits in receiving mode. 1 = Even number of logic 1’s are transmitted or check the data word and parity bits in receiving mode. **Note:** This bit has effect only when PBE bit (parity bit enable) is set.
[3]	**PBE**	**Parity Bit Enable Control** 0 = Parity bit is not generated (transmitting data) or checked (receiving data) during transfer. 1 = Parity bit is generated or checked bet“een the “last data”word “it” and ”stop bit” of the serial data.
[2]	**NSB**	**Number of STOP Bit Length** 0 = 1 “ STOP bit” is generated in the transmitted data. 1 = 1.5 “STOP bit” is generated in the transmitted data when 5-bit word length is selected, and 2 STOP bit” is generated when 6, 7 and 8 bits data length is selected.
[1:0]	**DATA_LEN**	**Data Length** 00 = Word length is 5-bit. 01 = Word length is 6-bit. 10 = Word length is 7-bit. 11 = Word length is 8-bit.

UART Interrupt Enable Register (UARTx_IER)

Register	Offset	R/W	Description	Reset Value
UART_IER	UARTx_BA+0x0C	R/W	UART Interrupt Enable Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							LIN_IE
7	6	5	4	3	2	1	0
ABAUD_IE	WAKE_IE	BUF_ERR_IE	RTO_IE	MODEM_IE	RLS_IE	THRE_IE	RDA_IE

Bits	Description	
[31:9]	Reserved	Reserved.
[8]	LIN_IE	**LIN Interrupt Enable Control** 0 = INT_LIN Disable. 1 = INT_LIN Enabled.
[7]	ABAUD_IE	**Auto-baud Rate Interrupt Enable Control** 0 = INT_ABAUD Disable. 1 = INT_ABAUD Enabled.
[6]	WAKE_IE	**Wake-up Interrupt Enable Control** 0 = INT_WAKE Disable. 1 = INT_WAKE Enabled.
[5]	BUF_ERR_IE	**Buffer Error Interrupt Enable Control** 0 = INT_BUT_ERR Disable. 1 = INT_BUF_ERR Enabled.
[4]	RTO_IE	**RX Time-out Interrupt Enable Control** 0 = INT_TOUT Disable. 1 = INT_TOUT Enabled.
[3]	MODEM_IE	**Modem Status Interrupt Enable Control** 0 = INT_MOS Disable. 1 = INT_MOS Enabled.
[2]	RLS_IE	**Receive Line Status Interrupt Enable Control** 0 = INT_RLS Disable. 1 = INT_RLS Enabled.
[1]	THRE_IE	**Transmit Holding Register Empty Interrupt Enable Control** 0 = INT_THRE Disable. 1 = INT_THRE Enabled.

Bits	Description	
[0]	**RDA_IE**	**Receive Data Available Interrupt Enable Control** 0 = INT_RDA Disable. 1 = INT_RDA Enabled.

UART Interrupt Status Control Register (UARTx_ISR)

Register	Offset	R/W	Description	Reset Value
UART_ISR	UARTx_BA+0x10	R/W	UART Interrupt Status Register.	0x0000_0002

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							LIN_IS
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ABAUD_IS	WAKE_IS	BUF_ERR_IS	RTO_IS	MODEM_IS	RLS_IS	THRE_IS	RDA_IS

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**LIN_IS**	**LIN Interrupt Status Flag (Read Only)** This bit is set when the LIN TX header transmitted, RX header received or the SIN does not equal SOUT and if LIN_IE(UART_IER[8]) is set then the LIN interrupt will be generated. 0 = No LIN interrupt is generated. 1 = LIN interrupt is generated. **Note1:** This bit is read only, but can be cleared by it by writing “1” to BIT_ERR_F (UART_TRSR[5]), LIN_TX_F (UART_TRSR[3]) or LIN_RX_F (UART_TRSR[4]). **Note2:** This bit is cleared when both the BIT_ERR_F, LIN_TX_F and LIN_RX_F are cleared.
[7]	**ABAUD_IS**	**Auto-baud Rate Interrupt Status Flag (Read Only)** This bit is set when auto-baud rate detection function finished or the auto-baud rate counter was overflow and if ABAUD_IE (UART_IER[7]) is set then the auto-baud rate interrupt will be generated. 0 = No Auto-Baud Rate interrupt is generated. 1 = Auto-Baud Rate interrupt is generated. **Note1:** This bit is read only, but can be cleared by it by writing “1” to ABAUD_TOUT_F (UART_TRSR[2]) or ABAUD_F (UART_TRSR[1]). **Note2:** This bit is cleared when both the ABAUD_TOUT_F and ABAUD_F are cleared.
[6]	**WAKE_IS**	**Wake-up Interrupt Status Flag (Read Only)** This bit is set in Power-down mode, the receiver received data or CTSn signal. If WAKE_IE (UART_IER[6]) is set then the wake-up interrupt will be generated. 0 = No Wake-Up interrupt is generated. 1 = Wake-Up interrupt is generated. **Note:** This bit is read only, but can be cleared by it by writing “1” to it.
[5]	**BUF_ERR_IS**	**Buffer Error Interrupt Status Flag (Read Only)** This bit is set when the TX or RX-FIFO overflowed. When BUF_ERR_IS is set, the transfer maybe not correct. If BUF_ERR_IE (UART_IER[5]) is set then the buffer error interrupt will be generated.

Bits	Description	
		0 = No Buffer error interrupt is generated. 1 = Buffer error interrupt is generated. **Note1:** This bit is read only, but can be cleared by it by writing “1” to TX_OVER_F (UART_FSR[8]) or RX_OVER_F (UART_FSR[0]). **Note2:** This bit is cleared when both the TX_OVER_F and RX_OVER_F are cleared.
[4]	**RTO_IS**	**RX Time-out Interrupt Status Flag (Read Only)** This bit is set when the RX-FIFO is not empty and no activities occur in the RX-FIFO and the time-out counter equal to TOIC. If RTO_IE (UART_IER[4]) is set then the tout interrupt will be generated. 0 = No RX Time-Out interrupt is generated. 1 = RX Time-Out interrupt is generated. **Note:** This bit is read only and user can read UART_RBR (RX is in active) to clear it.
[3]	**MODEM_IS**	**MODEM Interrupt Status Flag (Read Only)** This bit is set when the CTSn pin has state change (DCTSF = “1”). If MODEM_IEN (UART_IER[3]) is set then the modem interrupt will be generated. 0 = No MODEM interrupt is generated. 1 = MODEM interrupt is generated. **Note:** This bit is read only, but can be cleared by it by writing “1” to DCT_F (UART_MCSR[18]).
[2]	**RLS_IS**	**Receive Line Interrupt Status Flag (Read Only)** This bit is set when the RX received data has parity error (PE_F (UART_FSR[4])), framing error (FE_F (UART_FSR[5])), break error (BI_F (UART_FSR[6])) or RS-485 detect address byte (RS-485_ADDET_F (UART_TRSR[0])).If RLS_IE (UART_IER[2]) is set then the RLS interrupt will be generated. 0 = No Receive Line interrupt is generated. 1 = Receive Line interrupt is generated. **Note1:** This bit is read only, but can be cleared by it by writing “1” to BI_F (UART_FSR[6]), FE_F (UART_FSR[5]), PE_F (UART_FSR[4]) or RS-485_ADDET_F (UART_TRSR[0]). **Note2:** This bit is cleared when the BI_F, FE_F, PE_F and RS-485_ADDET_F are cleared.
[1]	**THRE_IS**	**Transmit Holding Register Empty Interrupt Flag (Read Only)** This bit is set when the last data of TX-FIFO is transferred to Transmitter Shift Register. If THRE_IEN (UART_IER[1]) is set that the THRE interrupt will be generated. 0 = No Transmit Holding register empty interrupt is generated. 1 = Transmit Holding register empty interrupt generated. **Note:** This bit is read only and it will be cleared when writing data into THR (TX-FIFO not empty).
[0]	**RDA_IS**	**Receive Data Available Interrupt Flag (Read Only)** When the number of bytes in the RX-FIFO equals the RFITL then the RDA_IF will be set. If RDA_IEN (UART_IER[0]) is set then the RDA interrupt will be generated. 0 = No Receive Data available interrupt is generated. 1 = Receive Data available interrupt is generated. **Note:** This bit is read only and it will be cleared when the number of unread bytes of RX-FIFO drops below the threshold level (RFITL).

UART Transfer Status Register (UARTx_TRSR)

Register	Offset	R/W	Description	Reset Value
UART_TRSR	UARTx_BA+0x14	R/W	UART Transfer State Status Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							LIN_RX_SYNC_ERR_F
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		BIT_ERR_F	LIN_RX_F	LIN_TX_F	ABAUD_TOUT_F	ABAUD_F	RS-485_ADDET_F

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**LIN_RX_SYNC_ERR_F**	**LIN RX SYNC Error Flag (Read Only)** This bit is set to logic “1” when LIN received incorrect SYNC field. User can choose the header by setting LIN_HEAD_SEL (UART_ALT_CTL[5:4]) register. 0 = No LIN Rx sync error is generated. 1 = LIN Rx sync error is generated. **Note:** This bit is read only, but can be cleared by writing “1” to LIN_RX_F.
[7:6]	**Reserved**	Reserved.
[5]	**BIT_ERR_F**	**Bit Error Detect Status Flag (Read Only)** At TX transfer state, hardware will monitoring the bus state, if the input pin (SIN) state is not equal to the output pin (SOUT) state, BIT_ERR_F will be set. When occur bit error, hardware will generate an interrupt to CPU (INT_LIN). 0 = No Bit error interrupt is generated. 1 = Bit error interrupt is generated. **Note1:** This bit is read only, but it can be cleared by writing “1” to it. **Note2:** This bit is only valid when enabling the bit error detection function (BIT_ERR_EN (UART_ALT_CTL[8]) = “1”).
[4]	**LIN_RX_F**	**LIN RX Interrupt Flag (Read Only)** This bit is set to logic “1” when received LIN header field. The header may be “break field” or “break field + sync field” or “break field + sync field + PID field”, and it can be choose by setting LIN_HEAD_SEL (UART_ALT_CTL[5:4]) register. 0 = No LIN Rx interrupt is generated. 1 = LIN Rx interrupt is generated. **Note:** This bit is read only, but can be cleared by writing “1” to it.
[3]	**LIN_TX_F**	**LIN TX Interrupt Flag (Read Only)** This bit is set to logic “1” when LIN transmitted header field. The header may be “break field” or “break field + sync field” or “break field + sync field + PID field”, it can be choose

Bits	Description	
		by setting LIN_HEAD_SEL (UART_ALT_CTL[5:4]) register. 0 = No LIN Tx interrupt is generated. 1 = LIN Tx interrupt is generated. **Note:** This bit is read only, but can be cleared by writing "1" to it.
[2]	**ABAUD_TOUT_F**	**Auto-baud Rate Time-out Interrupt (Read Only)** This bit is set to logic "1" in Auto-baud Rate Detect mode and the baud rate counter is overflow. 0 = No Auto-Baud Rate Time-Out interrupt is generated. 1 = Auto-Baud Rate Time-Out interrupt is generated. **Note:** This bit is read only, but can be cleared by writing "1" to it.
[1]	**ABAUD_F**	**Auto-baud Rate Interrupt (Read Only)** This bit is set to logic "1" when auto-baud rate detect function finished. 0 = No Auto- Baud Rate interrupt is generated. 1= Auto-Baud Rate interrupt is generated. **Note:** This bit is read only, but can be cleared by writing "1" to it.
[0]	**RS-485_** **ADDET_F**	**RS-485 Address Byte Detection Status Flag (Read Only)** This bit is set to logic "1" and set RS-485_ADD_EN (UART_ALT_CTL[19]) whenever in RS-485 mode the receiver detected any address byte character (bit 9 ='1') bit". This bit is reset whenever the CPU writes "1" to this bit. 0 = No RS-485 address detection interrupt is generated. 1 = RS-485 address detection interrupt is generated. **Note1:** This field is used for RS-485 mode. **Note2:** This bit is read only, but can be cleared by writing "1" to it.

UART FIFO Status Register (UART_FSR)

Register	Offset	R/W	Description	Reset Value
UART_FSR	UARTx_BA+0x18	R/W	UART FIFO State Status Register.	0x0000_0A02

31	30	29	28	27	26	25	24
Reserved			**TX_POINTER_F**				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			**RX_POINTER_F**				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**TE_F**	**TX_FULL_F**	**TX_EMPTY_F**	**TX_OVER_F**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**BI_F**	**FE_F**	**PE_F**	**Reserved**	**RX_FULL_F**	**RX_EMPTY_F**	**RX_OVER_F**

Bits	Description	
[31:28]	**Reserved**	Reserved.
[28:24]	**TX_POINTER_F**	**TX-fIFO Pointer (Read Only)** This field indicates the TX-FIFO Buffer Pointer. When CPU writes one byte data into UART_THR, TX_POINTER_F increases one. When one byte of TX-FIFO is transferred to Transmitter Shift Register, TX_POINTER_F decreases one.
[23:21]	**Reserved**	Reserved.
[20:16]	**RX_POINTER_F**	**RX-fIFO Pointer (Read Only)** This field indicates the RX-FIFO Buffer Pointer. When UART receives one byte from external device, RX_POINTER_F increases one. When one byte of RX-FIFO is read by CPU, RX_POINTER_F decreases one.
[15:12]	**Reserved**	Reserved.
[11]	**TE_F**	**Transmitter Empty Status Flag (Read Only)** This bit is set by hardware when TX FIFO (UA_THR) is empty and the STOP bit of the last byte has been transmitted. This bit is cleared automatically when TX FIFO is not empty or the last byte transmission has not completed. 0 = TX FIFO is not empty. 1 = TX FIFO is empty.
[10]	**TX_FULL_F**	**Transmitter FIFO Full (Read Only)** This bit indicates TX-FIFO full or not. This bit is set when TX_POINTER_F is equal to 16, otherwise is cleared by hardware. 0 = TX FIFO is not full. 1 = TX FIFO is full.
[9]	**TX_EMPTY_F**	**Transmitter FIFO Empty (Read Only)** This bit indicates TX-FIFO empty or not. When the last byte of TX-FIFO has been transferred to Transmitter Shift Register, hardware sets this bit high. It will be cleared when writing data into THR (TX-FIFO not empty).

Bits	Description	
		0 = TX FIFO is not empty. 1 = TX FIFO is empty.
[8]	**TX_OVER_F**	**TX Overflow Error Interrupt Status Flag (Read Only)** If TX-FIFO (UART_THR) is full, an additional write to UART_THR will cause this bit to logic “1”. 0 = TX FIFO is not overflow. 1 = TX FIFO is overflow. **Note:** This bit is read only, but it can be cleared by writing “1” to it.
[7]	**Reserved**	Reserved.
[6]	**BI_F**	**Break Status Flag (Read Only)** This bit is set to a logic “1” whenever the received data input(RX) is held in the “spacing state” (logic “0”) for longer than a full word transmission time (that is, the total time of “start bit” + data bits + parity + stop bits) and it is reset whenever the CPU writes “1” to this bit. 0 = No Break interrupt is generated. 1 = Break interrupt is generated. **Note:** This bit is read only, but it can be cleared by writing “1” to it.
[5]	**FE_F**	**Framing Error Status Flag (Read Only)** This bit is set to logic “1” whenever the received character does not have a valid “stop bit” (that is, the stop bit following the last data bit or parity bit is detected as a logic “0”), and it is reset whenever the CPU writes “1” to this bit. 0 = No framing error is generated. 1 = Framing error is generated. **Note:** This bit is read only, but it can be cleared by writing “1” to it.
[4]	**PE_F**	**Parity Error State Status Flag (Read Only)** This bit is set to logic “1” whenever the received character does not have a valid “parity bit”, and it is reset whenever the CPU writes “1” to this bit. 0 = No parity error is generated. 1 = Parity error is generated. **Note:** This bit is read only, but it can be cleared by writing “1” to it.
[3]	**Reserved**	Reserved.
[2]	**RX_FULL_F**	**Receiver FIFO Full (Read Only)** This bit initiates RX-FIFO full or not. This bit is set when RX_POINTER_F is equal to 16, otherwise is cleared by hardware. 0 = RX FIFO is not full. 1 = RX FIFO is full.
[1]	**RX_EMPTY_F**	**Receiver FIFO Empty (Read Only)** This bit initiate RX-FIFO empty or not. When the last byte of RX-FIFO has been read by CPU, hardware sets this bit high. It will be cleared when UART receives any new data. 0 = RX FIFO is not empty. 1 = RX FIFO is empty.
[0]	**RX_OVER_F**	**RX Overflow Error Status Flag (Read Only)** This bit is set when RX-FIFO overflow. If the number of bytes of received data is greater than RX-FIFO (UART_RBR) size, 16 bytes of UART0/UART1, this bit will be set. 0 = RX FIFO is not overflow.

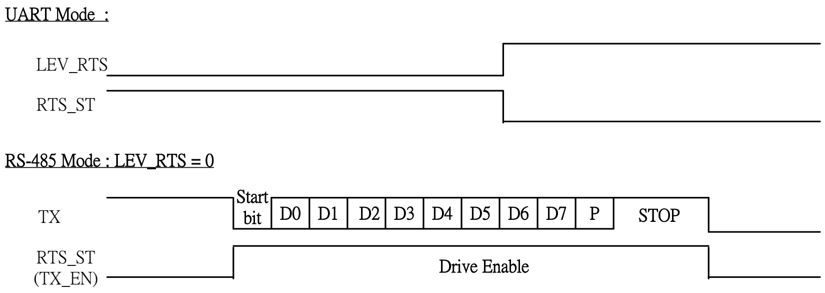
Bits	Description	
		1 = RX FIFO is overflow. **Note:** This bit is read only, but it can be cleared by writing “1” to it.

UART MODEM Control Register (UARTx_MCSR)

Register	Offset	R/W	Description	Reset Value
UART_MCSR	UARTx_BA+0x1C	R/W	UART Modem State Status Register.	0x0002_0002

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved					**DCT_F**	**CTS_ST**	**LEV_CTS**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**RTS_ST**	**LEV_RTS**

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:19]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
<tr><td>[18]</td><td><b>DCT_F</b></td><td><b>Detect CTSn State Change Status Flag (Read Only)</b>
This bit is set whenever CTSn input has change state, and it will generate Modem interrupt to CPU when MODEM_IE (UART_IER[3]).
0 = CTS input has not change state.
1 = CTS input has change state.
<b>Note:</b> This bit is read only, but it can be cleared by writing “1” to it.</td></tr>
<tr><td>[17]</td><td><b>CTS_ST</b></td><td><b>CTSn Pin Status (Read Only)</b>
This bit is the pin status of CTSn.
0 = CTS pin input is low level voltage logic state.
1 = CTS pin input is high level voltage logic state.</td></tr>
<tr><td>[16]</td><td><b>LEV_CTS</b></td><td><b>CTSn Trigger Level</b>
This bit can change the CTSn trigger level.
0 = Low level triggered.
1 = High level triggered.
<table>
<tr><th>Operation Mode</th><th>LEV_CTS</th><th>CTSn Pin Input</th><th>CTS_ST</th><th>Transmitter State</th></tr>
<tr><td rowspan="4">CTS Auto-Flow Control Mode</td><td>0</td><td>0</td><td>0</td><td>STOP</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>ACTIVE</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>ACTIVE</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>STOP</td></tr>
</table></td></tr>
<tr><td>[15:12]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
<tr><td>[1]</td><td><b>RTS_ST</b></td><td><b>RTSn Pin State (Read Only)</b>
This bit is the pin status of RTSn.</td></tr>
</table>

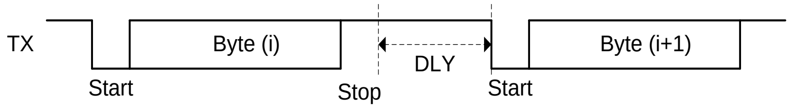
<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>0 = RTS pin input is low level voltage logic state.
1 = RTS pin input is high level voltage logic state.</td></tr>
<tr><td>[0]</td><td><b>LEV_RTS</b></td><td><b>RTSn Trigger Level</b>
This bit can change the RTSn trigger level.
0 = low level triggered.
1 = high level triggered.
For example, the relation waveform between LEV_RTS and RTSn shown as follows.
<table>
<tr><th>Operation Mode</th><th>LEV_RTS</th><th>RTS_ST (Default Output State)</th></tr>
<tr><td><b>RS-485 AUD Mode (Note)</b></td><td>0</td><td>0</td></tr>
<tr><td></td><td>1</td><td>1</td></tr>
<tr><td><b>RTS Auto-Flow Control Mode (Note)</b></td><td>0</td><td>1</td></tr>
<tr><td></td><td>1</td><td>0</td></tr>
<tr><td><b>Normal mode</b></td><td>0</td><td>1</td></tr>
<tr><td></td><td>1</td><td>0</td></tr>
</table>
<b>Note:</b> In RS-485 AUD mode and RTS Auto-flow control mode, hardware will control the output RTS pin automatically, so the table indicates the default value.


<b>Note:</b> The default setting in UART mode is LEV_RTS = “0” and RTS_ST = “1”.</td></tr>
</table>

UART TIME-OUT Register (UARTx_TMCTL)

Register	Offset	R/W	Description	Reset Value
UART_TMCTL	UARTx_BA+0x20	R/W	UART Time-Out Control State Register.	0x0000_01FF

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DLY							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**TOIC**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOIC							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**DLY**	**TX Delay Time Value** This field is used to programming the transfer delay time between the last stop bit and next start bit.  TX Byte (i) Byte (i+1) DLY Start Stop Start **Note1:** Fill all "0" to this field indicates to disable this function. **Note2:** The real delay value is DLY. **Note3:** The counting clock is baud rate clock.
[15:9]	**Reserved**	Reserved.
[8:0]	**TOIC**	**Time-out Comparator** The time-out counter resets and starts counting whenever the RX-FIFO receives a new data word. **Note1:** Fill all "0" to this field indicates to disable this function. **Note2:** The real time-out value is TOIC + 1. **Note3:** The counting clock is baud rate clock. **Note4:** The UART data format is start bit + 8 data bits + parity bit + stop bit, although software can configure this field by any value but it is recommend to filled this field great than 0xA.

UART Baud Rate Divider Register (UARTx_BAUD)

Register	Offset	R/W	Description	Reset Value
UART_BAUD	UARTx_BA+0x24	R/W	UART Baud Rate Divisor Register	0x0000_0000

31	30	29	28	27	26	25	24
DIV_16_EN	**Reserved**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BRD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BRD							

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31]</td><td><b>DIV_16_EN</b></td><td><b>Divider 16 Enable Control</b>
The BRD = Baud Rate Divider, and the baud rate equation is
Baud Rate = UART_CLK/ [M * (BRD + 1)]; The default value of M is 16.
0 = The equation of baud rate is UART_CLK / [(BRD+1)].
1 = The equation of baud rate is UART_CLK / [16 * (BRD+1)].
<b>Note:</b> In IrDA mode, this bit must clear to “0”.</td></tr>
<tr><td>[30:16]</td><td><b>Reserved</b></td><td>Reserved.</td></tr>
<tr><td>[15:0]</td><td><b>BRD</b></td><td><b>Baud Rate Divider</b>
The low byte of the baud rate divider<table><tr><th>DIV_16_EN</th><th>Baud Rate Equation</th></tr><tr><td>Disable
(Mode 0)</td><td>UART_CLK / (BRD+1), A must &gt;2</td></tr><tr><td>Enable
(Mode 1)</td><td>UART_CLK / [16 * (BRD+1)]</td></tr></table></td></tr>
</table>

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

UART IrDA Control Register (UARTx_IRCR)

Register	Offset	R/W	Description	Reset Value
UART_IRCR	UARTx_BA+0x30	R/W	UART IrDA Control Register.	0x0000_0040

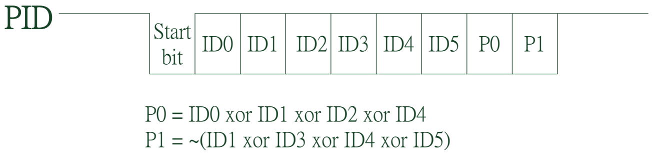
31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**INV_RX**	**INV_TX**	**Reserved**			**TX_SELECT**	**Reserved**

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**INV_RX**	**INV_RX** 0 = RX input signal no inversion. 1 = RX input signal inversion.
[5]	**INV_TX**	**INV_TX** 0 = TX output signal no inversion. 1 = TX output signal inversion.
[4:2]	**Reserved**	Reserved.
[1]	**TX_SELECT**	**TX_SELECT** 0 = Select IrDA receiver. 1 = Select IrDA transmitter.
[0]	**Reserved**	Reserved.

UART ALT Control State Register (UARTx_ALT_CTL)

Register	Offset	R/W	Description	Reset Value
UART_ALT_CSR	UARTx_BA+0x34	R/W	UART Alternate Control State Register.	0x0000_0000

31	30	29	28	27	26	25	24
ADDR_PID_MATCH							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				**RS-485_ ADD_EN**	**RS-485_AUD**	**RS-485_AAD**	**RS-485_NMM**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**BIT_ERR_EN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LIN_TX_EN	**LIN_RX_EN**	**LIN_HEAD_SEL**		**Reserved**	**LIN_TX_BCNT**		

Bits	Description	
[31:24]	**ADDR_PID_MATCH**	**Address / PID Match Value Register** When in the RS-485 Function Mode, this field contains the RS-485 address match values. When in the LIN Function mode, this field contains the LIN protected identifier field, software fills ID0~ID5 (ADDR_PID_MATCH [5:0]), hardware will calculate P0 and P1.  PID Start bit ID0 ID1 ID2 ID3 ID4 ID5 P0 P1 P0 = ID0 xor ID1 xor ID2 xor ID4 P1 = ~(ID1 xor ID3 xor ID4 xor ID5) **Note:** This field is used for RS-485 auto address detection mode or used for LIN protected identifier field (PID).
[23:20]	**Reserved**	Reserved.
[19]	**RS-485_ ADD_EN**	**RS-485 Address Detection Enable Control** This bit is used to enable RS-485 hardware address detection mode. 0 = Address detection mode Disabled. 1 = Address detection mode Enabled. **Note:** This field is used for RS-485 any operation mode.
[18]	**RS-485_AUD**	**RS-485 Auto Direction Mode (AUD Mode)** 0 = RS-485 Auto Direction mode (AUD) Disabled. 1 = RS-485 Auto Direction mode (AUD) Enabled. **Note:** It can be active in RS-485_AAD or RS-485_NMM operation mode.
[17]	**RS-485_AAD**	**RS-485 Auto Address Detection Operation Mode (AAD Mode)** 0 = RS-485 Auto Address Detection Operation mode (AAD) Disabled. 1 = RS-485 Auto Address Detection Operation mode (AAD) Enabled. **Note:** It can’t be active in RS-485_NMM Operation mode.

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

Bits	Description	
[16]	**RS-485_NMM**	**RS-485 Normal Multi-drop Operation Mode (NMM Mode)** 0 = RS-485 Normal Multi-drop Operation mode (NMM) Disabled. 1 = RS-485 Normal Multi-drop Operation mode (NMM) Enabled. **Note:** It can't be active in RS-485_AAD Operation mode.
[15:9]	**Reserved**	Reserved.
[8]	**BIT_ERR_EN**	**Bit Error Detect Enable Control** 0 = Bit error detection Disabled. 1 = Bit error detection Enabled. **Note:** In LIN function mode, when bit error occurs, hardware will generate an interrupt to CPU (INT_LIN).
[7]	**LIN_TX_EN**	**LIN TX Header Trigger Enable Control** 0 = LIN TX Header Trigger Disabled. 1 = LIN TX Header Trigger Enabled. **Note1:** This bit will be cleared automatically and generate a interrupt to CPU (INT_LIN). **Note2:** If user wants to receive transmit data, it recommended to enable LIN_RX_EN bit.
[6]	**LIN_RX_EN**	**LIN RX Enable Control** When LIN RX mode enabled and received a break field or sync field or PID field (Select by LIN_Header_SEL), the controller will generator a interrupt to CPU (INT_LIN) 0 = LIN RX mode Disabled. 1 = LIN RX mode Enabled.
[5:4]	**LIN_HEAD_SEL**	**LIN Header Selection** 00 = The LIN header includes "break field". 01 = The LIN header includes "break field + sync field". 10 = The LIN header includes "break field + sync field + PID field". 11 = Reserved.
[3]	**Reserved**	Reserved.
[2:0]	**LIN_TX_BCNT**	**LIN TX Break Field Count** The field contains 3-bit LIN TX break field count. **Note:** The break field length is LIN_TX_BCNT + 8.

UART Function Select Register (UARTx_FUN_SEL)

Register	Offset	R/W	Description	Reset Value
UART_FUN_SEL	UARTx_BA+0x38	R/W	UART Function Select Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						FUN_SEL	

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1:0]	**FUN_SEL**	**Function Select Enable Control** 00 = UART function mode. 01 = LIN function mode. 10 = IrDA function mode. 11 = RS-485 function mode.

UART Baud Rate Compensation Register (UARTx_BR_COMP)

Register	Offset	R/W	Description	Reset Value
UART_BR_COMP	UARTx_BA+0x3C	R/W	UART Baud Rate Compensation	0x0000_0000

31	30	29	28	27	26	25	24
BR_COMP_DEC	**Reserved**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**BR_COMP[8]**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BR_COMP[7:0]							

Bits	Description	
[31]	**BR_COMP_DEC**	**Baud Rate Compensation Decrease** 0 = Positive (increase one module clock) compensation for each compensated bit. 1 = Negative (decrease one module clock) compensation for each compensated bit.
[30:9]	**Reserved**	Reserved.
[8:0]	**BR_COMP**	**Baud Rate Compensation Patten** These 9-bits are used to define the relative bit is compensated or not. BR_COMP[7:0] is used to define the compensation of D[7:0] and BR_COM{[8] is used to define the parity bit.

6.15 Smart Card Host Interface (SC)

6.15.1 Overview

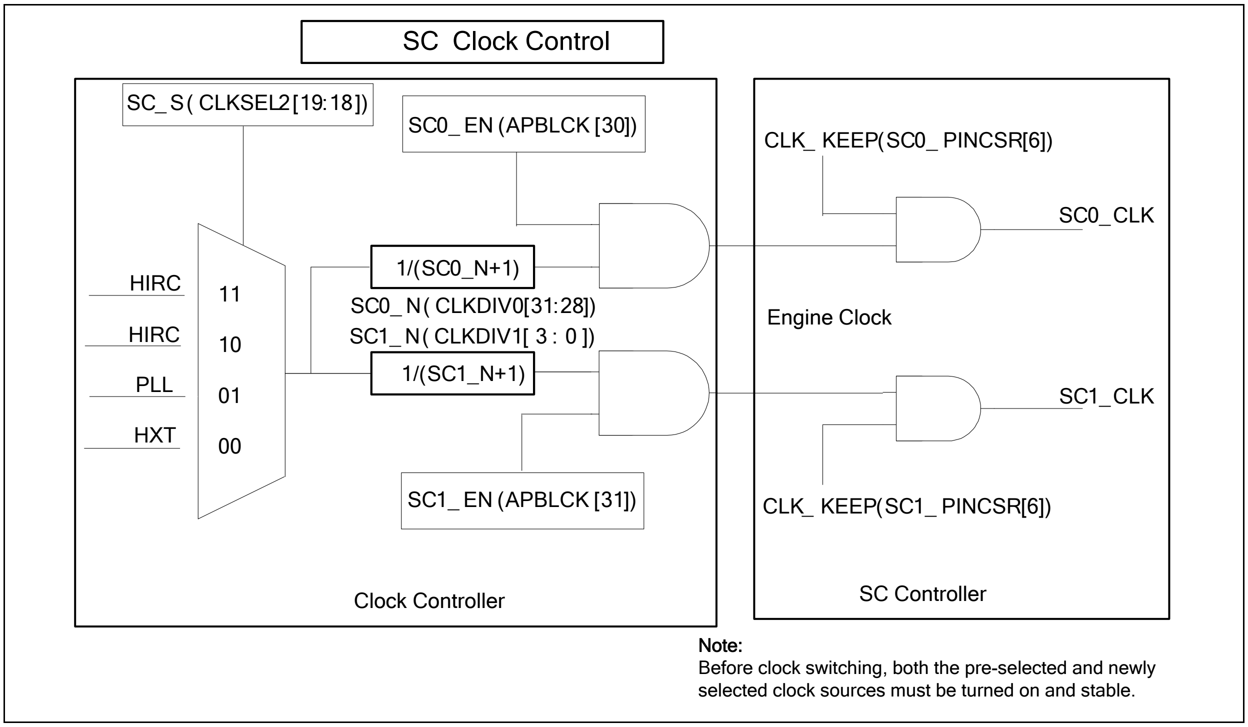
The Smart Card Interface controller (SC controller) is based on ISO/IEC 7816-3 standard and fully compliant with PC/SC Specifications. It also provides status of card insertion/removal.

6.15.2 Features

- ISO-7816-3 T = 0, T = 1 compliant.
- EMV2000 compliant
- Up to two ISO-7816-3 ports
- Separates receive/transmit 4 byte entry FIFO for data payloads.
- Programmable transmission clock frequency.
- Programmable receiver buffer trigger level.
- Programmable guard time selection (11 ETU ~ 267 ETU).
- A 24-bit and two 8 bit timers for Answer to Request (ATR) and waiting times processing.
- Supports auto inverse convention function.
- Supports transmitter and receiver error retry and error number limitation function.
- Supports hardware activation sequence process.
- Supports hardware warm reset sequence process.
- Supports hardware deactivation sequence process.
- Supports hardware auto deactivation sequence when detected the card removal.
- Supports UART mode
  - Full duplex, asynchronous communications.
  - Separates receiving / transmitting 4 bytes entry FIFO for data payloads.
  - Supports programmable baud rate generator for each channel.
  - Supports programmable receiver buffer trigger level.
  - Programmable transmitting data delay time between the last stop bit leaving the TX-FIFO and the de-assertion by setting SC_EGTR register.
  - Programmable even, odd or no parity bit generation and detection.
  - Programmable stop bit, 1 or 2 stop bit generation

6.15.3 Block Diagram

The SC clock control and block diagram are shown as follows. The SC controller is completely asynchronous design with to clock domains, PCLK and engine clock, note that the PCLK should be higher or equal than the frequency of engine clock.



Note:
Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.

Figure 6-66 SC Clock Control Diagram (4-bit Prescale Counter in Clock Controller)

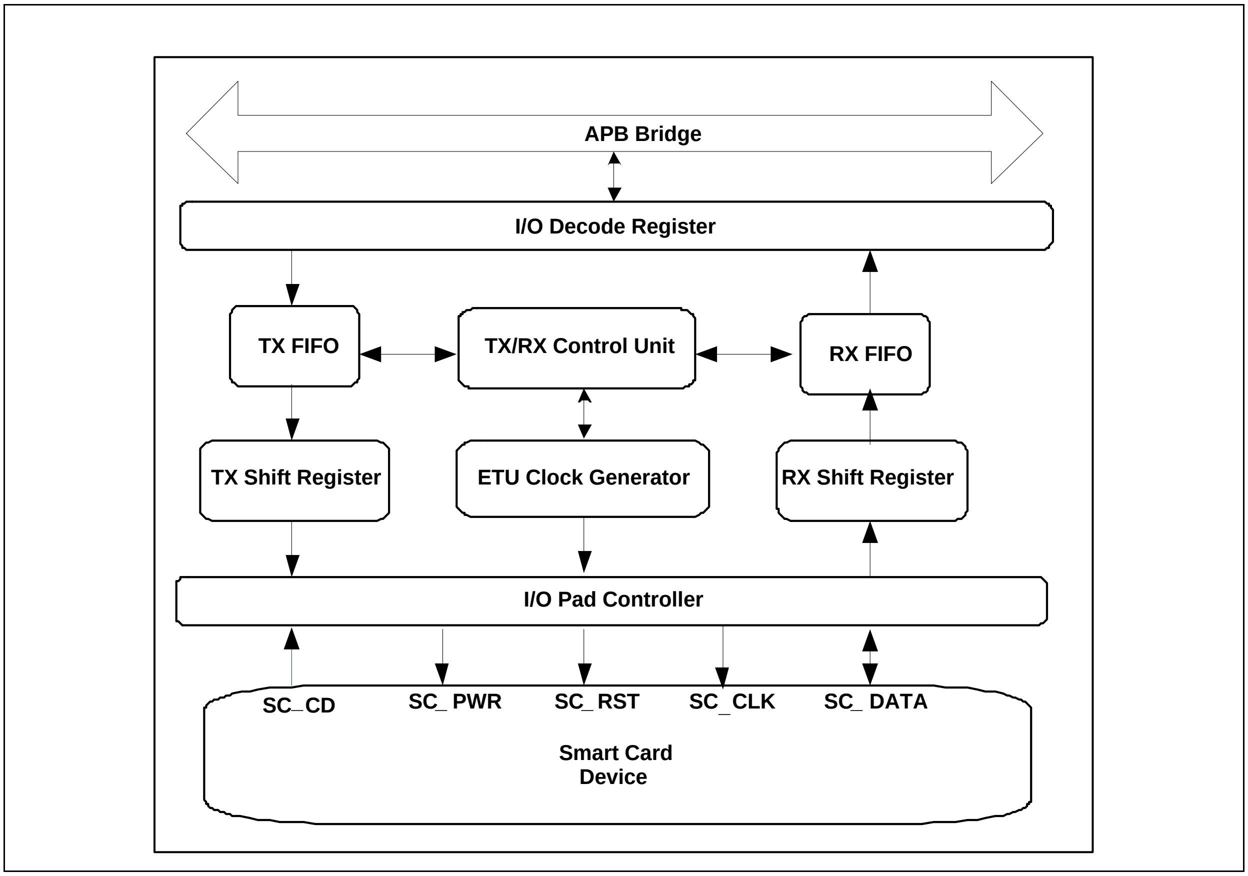


Figure 6-67 SC Controller Block Diagram

6.15.4 Functional description

Basically, the smart card interface acts as a half-duplex asynchronous communication port and its data format is composed of ten consecutive bits which is show as follows.

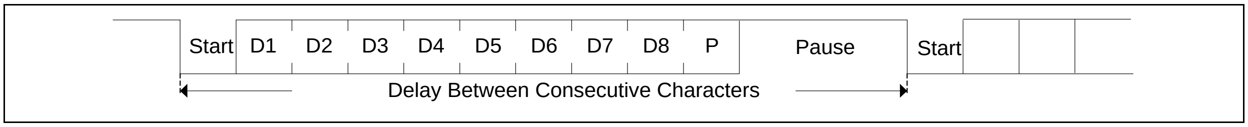


Figure 6-68 SC Data Character

6.15.4.1 Activation, Warm Reset and Deactivation Sequence

Activation

The Smart Card Interface controller supports hardware activation, warm reset and deactivation sequence. The activation sequence is show as Follows.

- Set SC_RST to low
- Set SC_PWR at high level and SC_DAT at high level (reception mode) .
- Enable SC_CLK clock

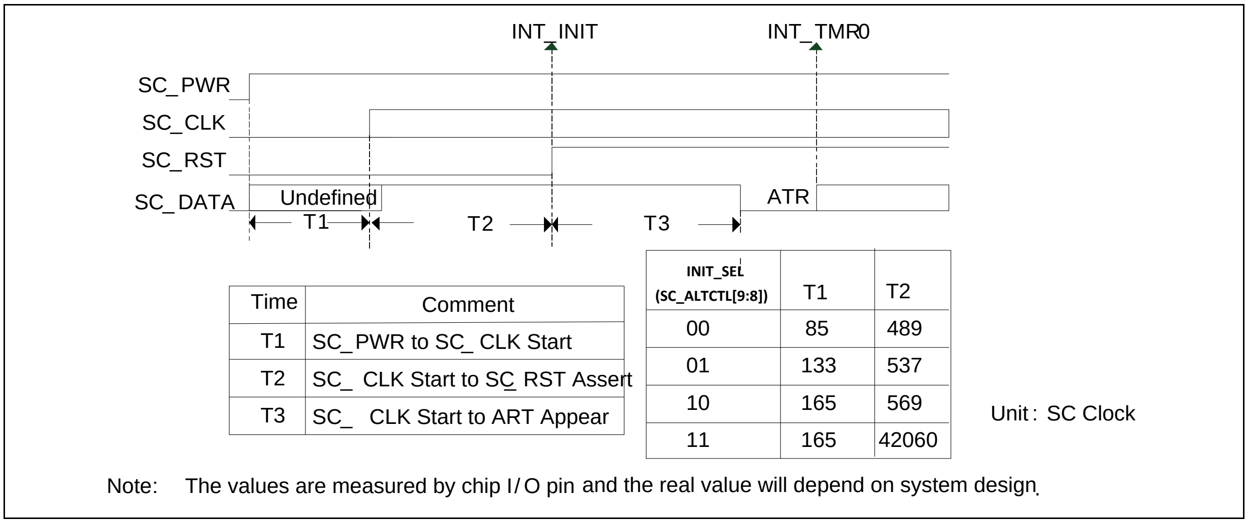
NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

- De-assert SC_RST to high

The activation sequence can be controlled by software or hardware. If software wants to control it, software can control SC_PINCSR and SC_TMRx register to process the activation sequence or setting ACT_EN(SC_ALTCTL[3]) register, and then the interface will perform the hardware activation sequence.

Following is activation control sequence in hardware activation mode:

- Set activation timing by setting INIT_SEL(SC_ALTCTL[9:8]).
- Timer0 can be selected when TMR_SEL(SC_CTL[14:13]) is 01, 10 or 11.
- Set operation mode MODE(SC_TMR0[27:24]) to 0011 and give an Answer to Request value by setting SC_TMR0 [CNT] register.
- When hardware de-asserts SC_RST to high, hardware will generator an interrupt INT_INIT to CPU at the same time INIT_IE(SC_IER[8]) = “1”
- If the Timer0 decreases the counter to “1” (start from SC_RST) and the card does not response ATR before that time, hardware will generate interrupt TMR0_IS (SC_ISR[3]) to CPU.



Time	Comment
T1	SC_PWR to SC_ CLK Start
T2	SC_ CLK Start to SC_RST Assert
T3	SC_ CLK Start to ART Appear

INIT_SEL (SC_ALTCTL[9:8])	T1	T2
00	85	489
01	133	537
10	165	569
11	165	42060

Note: The values are measured by chip I/O pin and the real value will depend on system design.

Figure 6-69 SC Activation Sequence

Warm Reset

The warm reset sequence is showed as follows.

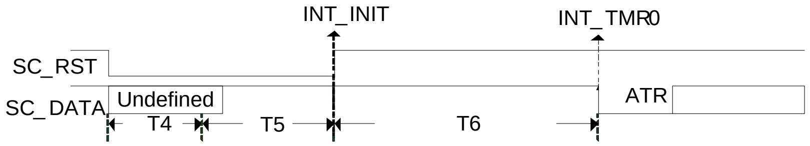
- Set SC_RST to low and set SC_DAT to high.
- Set SC_RST to high.

The warm reset sequence can be controlled by software or hardware. If software wants to control it, software can control SC_PINCSR and SC_TMRx register to process the warm reset sequence or set WARST_EN(SC_ALTCTL[4] register, and then the interface will perform the hardware warm reset sequence.

Following is warm reset control sequence in hardware warm reset mode

- Set warm reset timing by setting INIT_SEL(SC_ALTCTL[9:8]).
- Select Timer0 by setting TMR_SEL(SC_CTL[14:13]) register (TMR_SEL can be 01, 10, or 11).
- Set operation mode MODE(SC_TMR0[27:24]) to 011 and give an Answer to Request value by setting CNT(SC_TMR0[23:0]) register.

- Set TM0_SEN(SC_ALTCTL[5] and WARST_EN(SC_ALTCTL[4]) to start counting.
- When hardware de-asserts SC_RST to high, hardware will generate an interrupt INT_INIT to CPU at the same time (INIT_IE(SC_IER[8] = “1”)
- If the Timer0 decreases the counter to “1” (start from SC_RST) and the card does not response ATR before that time, hardware will generate interrupt INT_TMR0 to CPU.



Time	Comment
T4	SC_RST to SC_DATA Reception Mode
T5	SC_DATA Reception Mode to SC_RST Assert
T6	SC_CLK Start to ATR Appear

INIT_SEL (SC_ALTCTL[9:8])	T4	T5
00	81	483
01	129	531
10	161	563
11	161	42106

Unit : SC Clock

Note: This value is measured by chip IO pin, the real value will depended on system design

Figure 6-70 SC Warm Reset Sequence

Deactivation

The deactivation sequence is showed as follows:

- Set SC_RST to low.
- Stop SC_CLK.
- Set SC_DATA to state low.
- Deactivated SC_PWR.

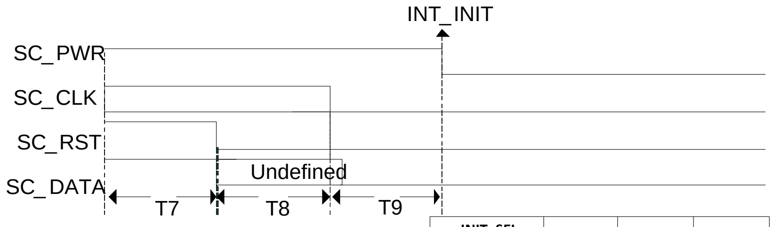
The deactivation sequence can be controlled by software or hardware. If software wants to control it, software can control SC_PINCSR and SC_TMR0 register to process the deactivation sequence or set DACT_EN(SC_ALTCTL[2]) register, and then the interface will perform the hardware deactivation sequence.

The SC controller also supports auto deactivation sequence when the card removal detection is set ADAC_CDEN(SC_ALTCTL[11])

Following is deactivation control sequence in hardware deactivation mode:

Set deactivation timing by setting INIT_SEL(SC_ALTCTL[9:8]). Set DACT_EN(SC_ALTCTL[2]) to start counting by SC_ALTCTL register.

When hardware de-asserts SC_PWR to low, the controller will generate an interrupt INT_INIT to CPU at the same time (if SC_IER[INIT_IE] = 1)



Time	Comment
T7	Deactivation Trigger to SC_RST Low
T8	SMC_ RST Low to Stop SC_CLK
T9	Stop SC_ CLK to Stop SC_PWR

INIT_SEL (SC_ALTCTL[9:8])	T7	T8	T9
00	97	83	87
01	145	131	135
10	177	163	167
11	177	163	167

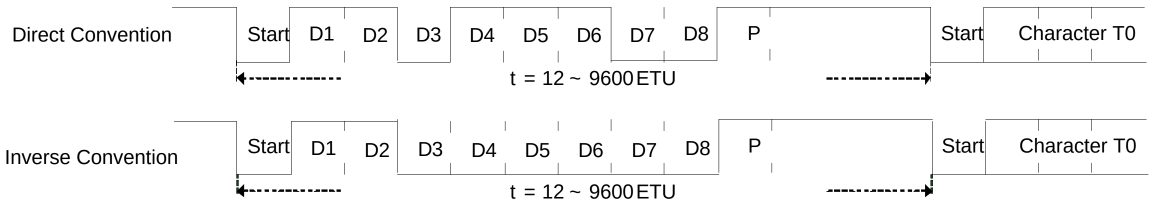
Unit: SC Clock

Figure 6-71 SC Deactivation Sequence

6.15.4.2 *Initial Character TS*

According to 7816-3, the initial character TS of answer to reset follows T=0 mode character description, request (ATR) has two possible patterns (as shown in the following figure). If the TS pattern is 1100_0000, it is inverse convention. When decoded by inverse convention, the conveyed byte is equal to 0x3F. If the TS pattern is 1101_1100, it is direct convention. When decoded by direct convention, the conveyed byte is equal to 0x3B.Software can set AUTO_CON_EN(SC_CTL[3]) and then the operating convention will be decided by hardware. Software can also set the CON_SEL(SC_CTL[5:4]) register (set to 00 or 11) to change the operating convention after SC received TS of answer to request (ATR).

If software enables auto convention function by setting AUTO_CON_EN(SC_CTL[3])register, the setting step must be done before Answer to Request state and the first data must be 0x3B or 0x3F. After hardware received first data and stored it at buffer, the hardware will decided the convention and change the CON_SEL(SC_CTL[5:4]) register automatically. If the first data is neither 0x3B nor 0x3F, the hardware will generate an interrupt INT_ACON_ERR (if ACON_ERR_IE (SC_IER [10])= “1”) to CPU.



Direct Convention	0_1101_1100_1(0x3B)
Inverse Convention	0_1100_0000_1(0x3F)

Figure 6-72 Initial Character TS

6.15.4.3 *Error Signal and Character Repetition*

According to ISO7816-3 T=0 mode description, as shown in following, if the receiver receives a wrong parity bit, it will pull the SC_DAT to low by 1.5 bit period to inform the transmitter parity error. Then the transmitter will retransmit the character. The SC interface controller supports hardware error detection function in receiver and supports hardware re-transmit function in transmitter. Software can enable re-transmit function by setting TX_ERETRY_EN(SC_CTL[23]). Software can also define the retry (re-transmit) number limitation in TX_ERETRY(SC_CTL[22:20]) register. The re-transmit number is up to TX_ERETRY +1 and if the re-transmit number is equal to TX_ERETRY +1, TX_OVER_REERR flag will be set by hardware and if TERR_IE (SC_IER [2], SC controller will generate a transfer error interrupt to CPU. Software can also define the received retry number limitation in RX_ERETRY(SC_CTL[18:16]) register. The receiver retry number is up to RX_ERETRY +1, if the number of received errors by receiver is equal to RX_ERETRY +1, receiver will receive this error data to buffer and RX_OVER_REERR flag will be set by hardware and if TERR_IE(SC_IER[2]), SC controller will generate a transfer error interrupt to CPU.

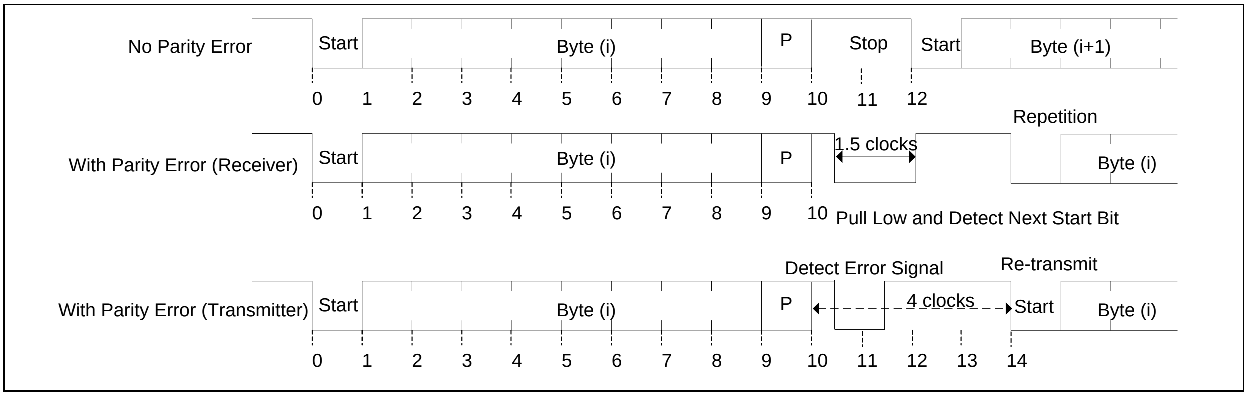


Figure 6-73 SC Error Signal

6.15.4.4 *Internal Time-out Counter*

The smart card interface includes a 24-bit time-out counter and two 8 bit time-out counters. These counters help the controller in processing different real-time interval (ATR, WWT, BWT, etc.). Each counter can be set to start counting once the trigger enable bit has been written or a START bit has been detected.

The following is the programming flow:

Enable counter by setting TMR_SEL (SC_CTL[14:13]). Select operation mode MODE (SC_TMRx[27:24]) and give a count value CNT(SC_TMRx[23:0]) by setting SC_TMRx register. Set TMR0_SEN (SC_ALTCTL [5], TMR1_SEN (SC_ALTCTL [6] or TMR2_SEN (SC_ALTCTL [7]) is to start counting.

The SC_TMR0, SC_TMR1 and SC_TMR2 timer operation mode are listed below table

Note: Only SC_TMR0 supports mode 0011.

TMRx **_SEL** **(X=0 ~2)**	**Operation Description**	
0000	The down counter started when TMRx_SEN (SC_ALTCTL[7:5]) enabled and ended when counter time-out. The time-out value will be CNT (SC_TMR0[32:0], SC_TMR1[7:0], SC_TMR2[7:0])+1	
	Start	Start counting when TMRx_SEN (SC_ALTCTL[7:5]) enabled
	End	When the down counter equals to 0, hardware will set TMRx_IS(SC_ISR[5:3]) and clear

		TMRx_SEN (SC_ALTCTL[7:5]) automatically.
0001	The down counter started when the first START bit (reception or transmission) detected and ended when counter time-out. The time-out value will be CNT (SC_TMR0[32:0], SC_TMR1[7:0], SC_TMR2[7:0])+1.	
	Start	Start counting when the first START bit (reception or transmission) detected after TMRx_SEN (SC_ALTCTL[7:5]) set to 1.
	End	When the down counter equals to 0, hardware will set TMRx_IS(SC_ISR[5:3]) and clear TMRx_SEN (SC_ALTCTL[7:5]) automatically.
0010	The down counter started when the first START bit (reception) detected and ended when counter time-out . The time-out value will be CNT (SC_TMR0[32:0], SC_TMR1[7:0], SC_TMR2[7:0])+1.	
	Start	Start counting when the first START bit (reception) detected bit after TMRx_SEN (SC_ALTCTL[7:5]) set to 1.
	End	When the down counter equals to 0, hardware will set TMRx_IS(SC_ISR[5:3]) and clear TMRx_SEN (SC_ALTCTL[7:5]) automatically.
0011	The down counter is only used for hardware activation, warm reset sequence to measure ATR timing. The timing starts when SC_RST de-assertion and ends when ATR response received or time-out. If the counter decreases to 0 before ATR response received, hardware will generate an interrupt to CPU. The time-out value will be CNT (SC_TMR0[32:0])+1.	
	Start	Start counting when SC_RST de-assertion after TMR0_SEN (SC_ALTCTL[5]) set to 1. It is used for hardware activation, warm reset mode.
	End	When the down counter equals to 0 before ATR response received, hardware will set TMR0_IS and clear TMR0_SEN (SC_ALTCTL[5]) automatically. When ATR received and down counter does not equal to 0, hardware will clear TMR0_SEN (SC_ALTCTL[5]) automatically.
0100	Same as 0000, but when the down counter equals to 0, hardware will set TMRx_IS(SC_ISR[5:3]) and counter will re-load the SC_TMRx [CNT] value and re-count until software clears TMRx_SEN (SC_ALTCTL[7:5]). When TMRx_ATV (SC_ALTCTL[15:13]) =1, software can change CNT (SC_TMR0[32:0], SC_TMR1[7:0], SC_TMR2[7:0]) value at any time. When the down counter equals to 0, counter will reload the new value of CNT (SC_TMR0[32:0], SC_TMR1[7:0], SC_TMR2[7:0]) and re-count. The time-out value will be CNT (SC_TMR0[32:0], SC_TMR1[7:0], SC_TMR2[7:0])+1.	
0101	Same as 0001, but when the down counter equals to 0, hardware will set TMRx_IS(SC_ISR[5:3]) and counter will re-load the CNT (SC_TMR0[32:0], SC_TMR1[7:0], SC_TMR2[7:0]) value. When the next START bit is detected, counter will re-count until software clears TMRx_ATV (SC_ALTCTL[15:13]). When TMRx_ATV (SC_ALTCTL[15:13]) =1 software can change CNT (SC_TMR0[32:0], SC_TMR0[7:0], SC_TMR0[7:0]) value at any time. When the down counter equal to 0, it will reload the new value of CNT (SC_TMR0[32:0], SC_TMR1[7:0], SC_TMR2[7:0]) and re-counting. The time-out value will be CNT (SC_TMR0[32:0], SC_TMR1[7:0], SC_TMR2[7:0])+1.	
0110	Same as 0010, but when the down counter equals to 0, it will set TMRx_IS(SC_ISR[5:3]) and counter will re-load the CNT (SC_TMR0[32:0], SC_TMR1[7:0], SC_TMR2[7:0]) value. When the next START bit is detected, counter will re-count until software clears TMRx_SEN (SC_ALTCTL[7:5]). When TMRx_ATV (SC_ALTCTL[15:13]) =1, software can change CNT (SC_TMR0[32:0], SC_TMR1[7:0], SC_TMR2[7:0]) value at any time. When the down counter equals to 0, counter will reload the new value of CNT (SC_TMR0[32:0], SC_TMR1[7:0], SC_TMR2[7:0]) and re-count. The time-out value will be CNT (SC_TMR0[32:0], SC_TMR1[7:0], SC_TMR2[7:0])+1.	
0111	The down counter started when the first START bit (reception or transmission) detected and ended when software clears TMRx_SEN (SC_ALTCTL[7:5]) bit. If next START bit detected, counter will reload the new value of CNT (SC_TMR0[32:0], SC_TMR1[7:0], SC_TMR2[7:0]) and re-counting. If the counter decreases to 0 before the next START bit detected, hardware will generate an interrupt to CPU. The time-out value will be CNT (SC_TMR0[32:0], SC_TMR1[7:0], SC_TMR2[7:0])+1.	
	Start	Start counting when the first START bit detected after TMRx_SEN (SC_ALTCTL[7:5]) set to 1.
	End	Stop counting after TMRx_SEN (SC_ALTCTL[7:5]) set to 0.

1000	The up counter starts when TMRx_SEN (SC_ALTCTL[7:5]) enabled and ends when TMRx_SEN (SC_ALTCTL[7:5]) disabled. This count value will be stored in TDRx(SC_TDRA [23:0], SC_TDRB[7:0], SC_T)R][1]:8). In this mode, hardware cannot generate any interrupt to CPU. The real count value will be TDRx(SC_TDRA [23:0], SC_TDRB[7:0], SC_T)R][1]:8) +1.	
	Start	Start counting after TMRx_SEN (SC_ALTCTL[7:5]) set to 1, and the start count value is 0 (hardware will ignore CNT (SC_TMR0[32:0], SC_TMR1[7:0], SC_TMR2[7:0]) value).
	End	Stop counting after TMRx_SEN (SC_ALTCTL[7:5]) set to 0 and the value stored to TDRx(SC_TDRA [23:0], SC_TDRB[7:0], SC_T)R][1]:8) register.
1111	Down counter starts when software set TMRx_SEN (SC_ALTCTL[7:5]) bit or any START bit been detected and ends when software clears TMRx_SEN (SC_ALTCTL[7:5]) bit. If next START bit detected, counter will reload the new value of CNT (SC_TMR0[32:0], SC_TMR1[7:0], SC_TMR2[7:0]) and re-counting. If the counter decreases to “0” before the next START bit be detected, hardware will generate an interrupt to CPU. The time-out value will be CNT (SC_TMR0[32:0], SC_TMR1[7:0], SC_TMR2[7:0])+1.	
	Start	Start count when the TMRx_SEN (SC_ALTCTL[7:5]) set to “1” or any START bit (TMRx_SEN (SC_ALTCTL[7:5]) must be set) be detected
	End	Stop count after TMRx_SEN (SC_ALTCTL[7:5]) set to “0”.

6.15.4.5 UART Mode

When the UA_MODE_EN(SC_UACTL[0]) bit set, the Smart Card Interface controller can also be used as base UART function. The following is the program example for UART mode.

Program example:

1. Software can entry UART mode by setting UA_MODE_EN(SC_UACTL[0]) bit.
2. Do software reset by setting RX_RST(SC_ALTCTL[1]) and TX_RST(SC_ALTCTL[0]) SCx_ALTCTL[TX_RST] bit to ensure that all state machine return idle state.
3. Filled “0” to CON_SEL(SC_CTL[5:4]) and AUTO_CON_EN (SC_CTL[3]) field. (In UART mode, those fields must be “0”)
4. Select the UART baud rate by setting ETU_RDIV (SC_ETUCR[11:0]) fields. For example, if smartcard module clock is 12 MHZ and target baudrate is 115200bps, ETU_RDIV should fill with (12000000 / 1–52–0 - 1).
5. Select the data format include data length (by setting DATA_LEN(SC_UACTL [5:4]), parity format (by setting OPE(SC_UACTL[7]) and PBDIS(SC_UACTL[6])) and stop bit length (by setting SLEN(SC_CTL[15] or EGT(SC_EGTR[7:0])).
6. Select the receiver buffer trigger level by setting RX_FTRI_LEV(SC_CTL[7:6]) field and select the receiver buffer time-out value by setting RFTMR (SC_RFTMR[8:0] field.
7. Write SC_THR (TX) register or read SC_RBR (RX) register can perform UART function.

6.15.5 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
SC Base Address: **SCx_BA = 0x4019_0000 + 0x20000*x** **x= 0, 1**				
SC_RBR	SCx_BA+0x00	R	SC Receiving Buffer Register.	0xXXXX_XXXX
SC_THR	SCx_BA+0x00	W	SC Transmit Holding Register.	0xXXXX_XXXX
SC_CTL	SCx_BA+0x04	R/W	SC Control Register.	0x0000_0000
SC_ALTCTL	SCx_BA+0x08	R/W	SC Alternate Control Register.	0x0000_0000
SC_EGTR	SCx_BA+0x0C	R/W	SC Extend Guard Time Register.	0x0000_0000
SC_RFTMR	SCx_BA+0x10	R/W	SC Receive buffer Time-out Register.	0x0000_0000
SC_ETUCR	SCx_BA+0x14	R/W	SC ETU Control Register.	0x0000_0173
SC_IER	SCx_BA+0x18	R/W	SC Interrupt Enable Control Register.	0x0000_0000
SC_ISR	SCx_BA+0x1C	R/W	SC Interrupt Status Register.	0x0000_0002
SC_TRSR	SCx_BA+0x20	R/W	SC Status Register.	0x0000_0202
SC_PINCSR	SCx_BA+0x24	R/W	SC Pin Control State Register.	0x0000_00x0
SC_TMR0	SCx_BA+0x28	R/W	SC Internal Timer Control Register 0.	0x0000_0000
SC_TMR1	SCx_BA+0x2C	R/W	SC Internal Timer Control Register 1.	0x0000_0000
SC_TMR2	SCx_BA+0x30	R/W	SC Internal Timer Control Register 2.	0x0000_0000
SC_UACTL	SCx_BA + 0x34	R/W	SC UART Mode Control Register.	0x0000_0000
SC_TDRA	SCx_BA+0x38	R	SC Timer Current Data Register A.	0x0000_07FF
SC_TDRB	SCx_BA+0x3C	R	SC Timer Current Data Register B.	0x0000_7F7F

6.15.6 Register Description

SC Receiving Buffer Register (SC_RBR)

Register	Offset	R/W	Description	Reset Value
SC_RBR	SCx_BA+0x00	R	SC Receiving Buffer Register.	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RBR							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**RBR**	**Receiving Buffer** By reading RBR, the SC will return an 8-bit received data.

SC Transmit Holding Register (SC_THR)

Register	Offset	R/W	Description	Reset Value
SC_THR	SCx_BA+0x00	W	SC Transmit Holding Register.	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
THR							

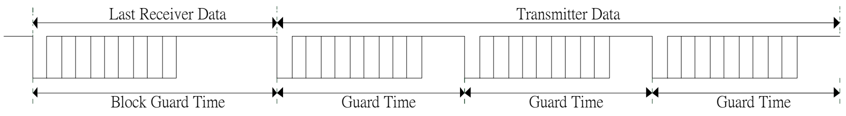
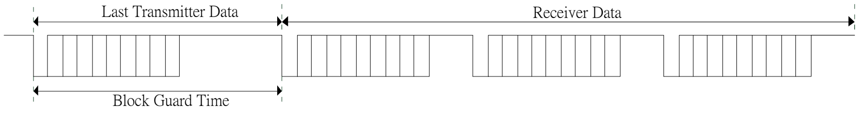
Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**THR**	**Transmit Holding Buffer** By writing data to THR, the SC will send out an 8-bit data. **Note:** If SC_CEN(SC_CTL[0]) is not enabled, THR cannot be programmed.

SC Control Register (SC_CTL)

Register	Offset	R/W	Description	Reset Value
SC_CTL	SCx_BA+0x04	R/W	SC Control Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						**CD_DEB_SEL**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TX_ERETRY_EN	**TX_ERETRY**			**RX_ERETRY_EN**	**RX_ERETRY**		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SLEN	**TMR_SEL**		**BGT**				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RX_FTRI_LEV		**CON_SEL**		**AUTO_CON_EN**	**DIS_TX**	**DIS_RX**	**SC_CEN**

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25:24]	**CD_DEB_SEL**	**Card Detect De-bounce Selection** This field indicates the card detect de-bounce selection. 00 = De-bounce sample card insert once per 384 (128 * 3) engine clocks and de-bounce sample card removal once per 128 engine clocks. 01 = De-bounce sample card insert once per 192 (64 * 3) engine clocks and de-bounce sample card removal once per 64 engine clocks. 10 = De-bounce sample card insert once per 96 (32 * 3) engine clocks and de-bounce sample card removal once per 32 engine clocks. 11 = De-bounce sample card insert once per 48 (16 * 3) engine clocks and de-bounce sample card removal once per 16 engine clocks.
[23]	**TX_ERETRY_EN**	**TX Error Retry Enable Control** This bit enables transmitter retry function when parity error has occurred. 0 = TX error retry function Disabled. 1 = TX error retry function Enabled.
[22:20]	**TX_ERETRY**	**TX Error Retry Count Number** This field indicates the maximum number of transmitter retries that are allowed when parity error has occurred. **Note1:** The real retry number is TX_ERETRY + 1, so 8 is the maximum retry number. **Note2:** This field cannot be changed when TX_ERETRY_EN enabled. The change flow is to disable TX_ETRTRY_EN first and then fill in new retry value.

[19]	**RX_ERETRY_EN**	**RX Error Retry Enable Control** This bit enables receiver retry function when parity error has occurred. 0 = RX error retry function Disabled. 1 = RX error retry function Enabled. **Note:** Software must fill in the RX_ERETRY value before enabling this bit.
[18:16]	**RX_ERETRY**	**RX Error Retry Count Number** This field indicates the maximum number of receiver retries that are allowed when parity error has occurred **Note1:** The real retry number is RX_ERETRY + 1, so 8 is the maximum retry number. **Note2:** This field cannot be changed when RX_ERETRY_EN enabled. The change flow is to disable RX_ETRTRY_EN first and then fill in new retry value.
[15]	**SLEN**	**Stop Bit Length** This field indicates the length of stop bit. 0 = The stop bit length is 2 ETU. 1= The stop bit length is 1 ETU. **Note:** The default stop bit length is 2. SMC and UART adopts SLEN to program the stop bit length
[14:13]	**TMR_SEL**	**Timer Selection** 00 = All internal timer function Disabled. 01 = Internal 24 bit timer Enabled. Software can configure it by setting SC_TMR0 [23:0]. SC_TMR1 and SC_TMR2 will be ignored in this mode. 10 = internal 24 bit timer and 8 bit internal timer Enabled. Software can configure the 24 bit timer by setting SC_TMR0 [23:0] and configure the 8 bit timer by setting SC_TMR1[7:0]. SC_TMR2 will be ignored in this mode. 11 = Internal 24 bit timer and two 8 bit timers Enabled. Software can configure them by setting SC_TMR0 [23:0], SC_TMR1 [7:0] and SC_TMR2 [7:0].
[12:8]	**BGT**	**Block Guard Time (BGT)** Block guard time means the minimum bit length between the leading edges of two consecutive characters between different transfer directions. This field indicates the counter for the bit length of block guard time. According to ISO7816-3, in T = 0 mode, software must fill 15 (real block guard time = 16.5) to this field; in T = 1 mode, software must fill 21 (real block guard time = 22.5) to it.  Last Receiver Data Transmitter Data Block Guard Time Guard Time Guard Time Guard Time Note1 : Hardware will control the transmit block guard time by BGT(SC_CTL[11:8]) register setting Note2 : Hardware will control the transmit guard time by EGT(SC_EGTR[7:0] register setting In RX mode, software can enable SC_ALTCTL [RX_BGT_EN] to detect the first coming character timing. If the incoming data timing less than BGT, an interrupt will be generated.  Note :If the incoming data timing less than SC_CTL [BGT], an interrupt will be generated (RX_BGT_EN(SC_ALTCTL[12]) enable

		Note: The real block guard time is BGT + 1.
[7:6]	**RX_FTRI_LEV**	**Rx Buffer Trigger Level** When the number of bytes in the receiving buffer equals the RX_FTRI_LEV, the RDA_IF will be set (if IER [RDA_IEN] is enabled, an interrupt will be generated). 00 = INTR_RDA Trigger Level with 01 byte. 01 = INTR_RDA Trigger Level with 02 bytes. 10 = INTR_RDA Trigger Level with 03 bytes. 11 = Reserved.
[5:4]	**CON_SEL**	**Convention Selection** 00 = Direct convention. 01 = Reserved. 10 = Reserved. 11 = Inverse convention. **Note:** If AUTO_CON_EN(SC_CTL[3]) enabled, this fields are ignored.
[3]	**AUTO_CON_EN**	**Auto Convention Enable Control** 0 = Auto-convention Disabled. 1 = Auto-convention Enabled. When hardware receives TS in answer to reset state and the TS is direct convention, CON_SEL(SC_CTL[5:4]) will be set to 00 automatically, otherwise if the TS is inverse convention, and CON_SEL (SC_CTL[5:4]) will be set to 11. If software enables auto convention function, the setting step must be done before Answer to Reset state and the first data must be 0x3B or 0x3F. After hardware received first data and stored it at buffer, hardware will decided the convention and change the CON_SEL (SC_CTL[5:4]) bits automatically. If the first data is not 0x3B or 0x3F, hardware will generate an interrupt INT_ACON_ERR (if ACON_ERR IE (SC_IER[10]) = 1 to CPU.
[2]	**DIS_TX**	**TX Transition Disable Control** 0 = The transceiver Enabled. 1 = The transceiver Disabled.
[1]	**DIS_RX**	**RX Transition Disable Control** 0 = The receiver Enabled. 1 = The receiver Disabled. **Note:** If AUTO_CON_EN is enabled, these fields must be ignored.
[0]	**SC_CEN**	**SC Engine Enable Control** Set this bit to 1 to enable SC operation. If this bit is cleared, SC will force all transition to IDLE state

SC Alternate Control Register (SC_ALTCTL)

Register	Offset	R/W	Description	Reset Value
SC_ALTCTL	SCx_BA+0x08	R/W	SC Alternate Control Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							OUTSEL
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TMR2_ATV	TMR1_ATV	TMR0_ATV	RX_BGT_EN	Reserved		INIT_SEL	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TMR2_SEN	TMR1_SEN	TMR0_SEN	WARST_EN	ACT_EN	DACT_EN	RX_RST	TX_RST

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**OUTSEL**	**Smartcard Data Pin Output Mode Selection** Use this bit to select smartcard data pin output mode 0 = Quasi mode. 1 = Open-drain mode.
[15]	**TMR2_ATV**	**Internal Timer2 Active State (Read Only)** This bit indicates the timer counter status of timer2. 0 = Timer2 is not active. 1 = Timer2 is active.
[14]	**TMR1_ATV**	**Internal Timer1 Active State (Read Only)** This bit indicates the timer counter status of timer1. 0 = Timer1 is not active. 1 = Timer1 is active.
[13]	**TMR0_ATV**	**Internal Timer0 Active State (Read Only)** This bit indicates the timer counter status of timer0. 0 = Timer0 is not active. 1 = Timer0 is active.
[12]	**RX_BGT_EN**	**Receiver Block Guard Time Function Enable Control** 0 = Receiver block guard time function Disabled. 1 = Receiver block guard time function Enabled.
[11:10]	**Reserved**	Reserved.
[9:8]	**INIT_SEL**	**Initial Timing Selection** This fields indicates the timing of hardware initial state (activation or warm-reset or deactivation). Unit: SC clock Activation: Refer to SC Activation Sequence in Figure 6‑69.

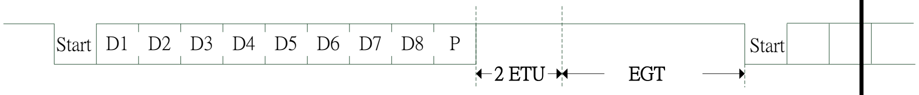
		Warm-reset: refer to Warm-Reset Sequence in Figure 6-70 Deactivation: refer to Deactivation Sequence in Figure 6-71
[7]	**TMR2_SEN**	**Internal Timer2 Start Enable Control** This bit enables Timer 2 to start counting. Software can fill 0 to stop it and set 1 to reload and count. 0 = Stops counting. 1 = Start counting. **Note1:** This field is used for internal 8 bit timer when TMR_SEL(SC_CTL[14:13]) = 11. Don't filled TMR2_SEN when TMR_SEL(SC_CTL[14:13]) = 00 or TMR_SEL(SC_CTL[14:13]) = 01 or TMR_SEL(SC_CTL[14:13]) = 10. **Note2:** If the operation mode is not in auto-reload mode (SC_TMR2[26] = 0), this bit will be auto-cleared by hardware. **Note3:** This field will be cleared by TX_RST(SC_ALTCTL[0]) and RX_RST(SC_ALTCTL[1]). So don't fill this bit, TX_RST(SC_ALTCTL[0]), and RX_RST(SC_ALTCTL[1]) at the same time. **Note4:** If SC_CEN(SC_CTL[0]) is not enabled, this filed cannot be programmed.
[6]	**TMR1_SEN**	**Internal Timer1 Start Enable Control** This bit enables Timer 1 to start counting. Software can fill 0 to stop it and set 1 to reload and count. 0 = Stops counting. 1 = Start counting. **Note1:** This field is used for internal 8 bit timer when TMR_SEL(SC_CTL[14:13]) = 10 or TMR_SEL(SC_CTL[14:13]) = 11. Don't filled TMR1_SEN when SC_CTL([TMR_SEL] = 00 or SC_CTL[TMR_SEL] = 01. **Note2:** If the operation mode is not in auto-reload mode (SC_TMR1[26] = 0), this bit will be auto-cleared by hardware. **Note3:** This field will be cleared by TX_RST(SC_ALTCTL[0]) and RX_RST(SC_ALTCTL[1]), so don't fill this bit, TX_RST(SC_ALTCTL[0]), and RX_RST(SC_ALTCTL[1]) at the same time. **Note4:** If SC_CEN(SC_CTL[0]) is not enabled, this filed cannot be programmed.
[5]	**TMR0_SEN**	**Internal Timer0 Start Enable Control** This bit enables Timer 0 to start counting. Software can fill 0 to stop it and set 1 to reload and count. 0 = Stops counting. 1 = Start counting. **Note1:** This field is used for internal 24 bit timer when TMR_SEL (SC_CTL[14:13]) = 01. **Note2:** If the operation mode is not in auto-reload mode (SC_TMR0[26] = 0), this bit will be auto-cleared by hardware. **Note3:** This field will be cleared by TX_RST(SC_ALTCTL[0]) and RX_RST(SC_ALTCTL[1]). So don't fill this bit, TX_RST and RX_RST at the same time. **Note4:** If SC_CEN(SC_CTL[0]) is not enabled, this filed cannot be programmed.
[4]	**WARST_EN**	**Warm Reset Sequence Generator Enable Control** This bit enables SC controller to initiate the card by warm reset sequence 0 = No effect. 1 = Warm reset sequence generator Enabled. **Note1:** When the warm reset sequence completed, this bit will be cleared automatically and the INIT_IS(SC_ISR[8]) will be set to 1. **Note2:** This field will be cleared by TX_RST(SC_ALTCTL[0]) and RX_RST(SC_ALTCTL[1]), so don't fill this bit, TX_RST, and RX_RST at the same time. **Note3:** If SC_CEN(SC_CTL[0]) is not enabled, this filed cannot be programmed
[3]	**ACT_EN**	**Activation Sequence Generator Enable Control** This bit enables SC controller to initiate the card by activation sequence

		0 = No effect. 1 = Activation sequence generator Enabled. **Note1:** When the activation sequence completed, this bit will be cleared automatically and the INIT_IS(SC_ISR[8]) will be set to 1. **Note2:** This field will be cleared by TX_RST(SC_ALTCTL[0]) and RX_RST(SC_ALTCTL[1]), so don't fill this bit, TX_RST(SC_ALTCTL[0]), and RX_RST(SC_ALTCTL[1]) at the same time. **Note3:** If SC_CEN(SC_CTL[0]) is not enabled, this filed cannot be programmed.
[2]	**DACT_EN**	**Deactivation Sequence Generator Enable Control** This bit enables SC controller to initiate the card by deactivation sequence 0 = No effect. 1 = Deactivation sequence generator Enabled. **Note1:** When the deactivation sequence completed, this bit will be cleared automatically and the INIT_IS(SC_ISR[8]) will be set to 1. **Note2:** This field will be cleared by TX_RST(SC_ALTCTL[0]) and RX_RST(SC_ALTCTL[1]). So don't fill this bit, TX_RST, and RX_RST at the same time. **Note3:** If SC_CTL [SC_CEN] not enabled, this filed cannot be programmed.
[1]	**RX_RST**	**Rx Software Reset** When RX_RST is set, all the bytes in the receiver buffer and Rx internal state machine will be cleared. 0 = No effect. 1 = Reset the Rx internal state machine and pointers. **Note:** This bit will be auto cleared after reset is complete.
[0]	**TX_RST**	**TX Software Reset** When TX_RST is set, all the bytes in the transmit buffer and TX internal state machine will be cleared. 0 = No effect. 1 = Reset the TX internal state machine and pointers. **Note:** This bit will be auto cleared after reset is complete.

SC Extend Guard Time Register (SC_EGTR)

Register	Offset	R/W	Description	Reset Value
SC_EGTR	SCx_BA+0x0C	R/W	SC Extend Guard Time Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EGT							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**EGT**	**Extended Guard Time** This field indicates the extended guard timer value.  Start D1 D2 D3 D4 D5 D6 D7 D8 P 2 ETU EGT Start **Note:** The counter is ETU base and the real extended guard time is EGT.

SC Receiver buffer Time-out Register (SC_RFTMR)

Register	Offset	R/W	Description	Reset Value
SC_RFTMR	SCx_BA+0x10	R/W	SC Receive buffer Time-out Register.	0x0000_0000

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th></tr>
<tr><td colspan="7">Reserved</td><td>RFTM</td></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">RFTM</td></tr>
</table>

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8:0]	**RFTM**	**SC Receiver Buffer Time-out (ETU Base)** The time-out counter resets and starts counting whenever the RX buffer received a new data word. Once the counter decrease to 1 and no new data is received or CPU does not read data by reading SC_RBR buffer, a receiver time-out interrupt INT_RTMR will be generated(if RTMR_IE(SC_IER[9]) = 1). **Note1: The counter unit is ETU based and the interval of time-out is RFTM + 0.5** **Note2:** Fill all 0 to this field indicates to disable this function.

SC Clock Divider Control Register (SC_ETUCR)

Register	Offset	R/W	Description	Reset Value
SC_ETUCR	SCx_BA+0x14	R/W	SC ETU Control Register.	0x0000_0173

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
COMPEN_EN	**Reserved**			**ETU_RDIV**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ETU_RDIV							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15]	**COMPEN_EN**	**Compensation Mode Enable Control** This bit enables clock compensation function. When this bit enabled, hardware will alternate between n clock cycles and n-1 clock cycles, where n is the value to be written into the ETU_RDIV . 0 = Compensation function Disabled. 1 = Compensation function Enabled.
[14:12]	**Reserved**	Reserved.
[11:0]	**ETU_RDIV**	**ETU Rate Divider** The field indicates the clock rate divider. The real ETU is ETU_RDIV + 1. **Note:** Software can configure this field, but this field must be greater than 0x004.

SC Interrupt Control Register (SC_IER)

Register	Offset	R/W	Description	Reset Value
SC_IER	SCx_BA+0x18	R/W	SC Interrupt Enable Control Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**ACON_ERR_IE**	**RTMR_IE**	**INIT_IE**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CD_IE	**BGT_IE**	**TMR2_IE**	**TMR1_IE**	**TMR0_IE**	**TERR_IE**	**TXBE_IE**	**RDA_IE**

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10]	**ACON_ERR_IE**	**Auto Convention Error Interrupt Enable Control** This field is used for auto-convention error interrupt enable. 0 = Auto-convention error interrupt Disabled. 1 = Auto-convention error interrupt Enabled.
[9]	**RTMR_IE**	**Receiver Buffer Time-out Interrupt Enable Control** This field is used for receiver buffer time-out interrupt enable. 0 = Receiver buffer time-out interrupt Disabled. 1 = Receiver buffer time-out interrupt Enabled.
[8]	**INIT_IE**	**Initial End Interrupt Enable Control** This field is used for activation (ACT_EN(SC_ALTCTL[3] = 1)), deactivation (DACT_EN SC_ALTCTL[2] = 1) and warm reset (SC_ALTCTL [WARST_EN]) sequence interrupt enable. 0 = Initial end interrupt Disabled. 1 = Initial end interrupt Enabled.
[7]	**CD_IE**	**Card Detect Interrupt Enable Control** This field is used for card detect interrupt enable. The card detect status is CD_INS_F(SC_SR[12]) 0 = Card detect interrupt Disabled. 1 = Card detect interrupt Enabled.
[6]	**BGT_IE**	**Block Guard Time Interrupt Enable Control** This field is used for block guard time interrupt enable. 0 = Block guard time Disabled. 1 = Block guard time Enabled.
[5]	**TMR2_IE**	**Timer2 Interrupt Enable Control** This field is used for TMR2 interrupt enable. 0 = Timer2 interrupt Disabled.

		1 = Timer2 interrupt Enabled.
[4]	**TMR1_IE**	**Timer1 Interrupt Enable Control** This field is used to enable the TMR1 interrupt. 0 = Timer1 interrupt Disabled. 1 = Timer1 interrupt Enabled.
[3]	**TMR0_IE**	**Timer0 Interrupt Enable Control** This field is used to enable TMR0 interrupt enable. 0 = Timer0 interrupt Disabled. 1 = Timer0 interrupt Enabled.
[2]	**TERR_IE**	**Transfer Error Interrupt Enable Control** This field is used for transfer error interrupt enable. The transfer error states is at SC_SR register which includes receiver break error RX_EBR_F(SC_SR[6]), frame error RX_EFR_F(SC_SR[5]), parity error RX_EPA_F(SC_SR[4]), receiver buffer overflow error RX_OVER_F(SC_SR[0]), transmit buffer overflow error TX_OVER_F(SC_SR[8]), receiver retry over limit error RX_OVER_REERR(SC_SR[22] and transmitter retry over limit error TX_OVER_REERR(SC_SR[30]. 0 = Transfer error interrupt Disabled. 1 = Transfer error interrupt Enabled.
[1]	**TXBE_IE**	**Transmit Buffer Empty Interrupt Enable Control** This field is used for transmit buffer empty interrupt enable. 0 = Transmit buffer empty interrupt Disabled. 1 = Transmit buffer empty interrupt Enabled.
[0]	**RDA_IE**	**Receive Data Reach Interrupt Enable Control** This field is used for received data reaching trigger level RX_FTRI_LEV (SC_CTL[7:6]) interrupt enable. 0 = Receive data reach trigger level interrupt Disabled. 1 = Receive data reach trigger level interrupt Enabled.

SC Interrupt Status Register (SC_ISR)

Register	Offset	R/W	Description	Reset Value
SC_ISR	SCx_BA+0x1C	R/W	SC Interrupt Status Register.	0x0000_0002

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**ACON_ERR_IS**	**RTMR_IS**	**INIT_IS**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CD_IS	**BGT_IS**	**TMR2_IS**	**TMR1_IS**	**TMR0_IS**	**TERR_IS**	**TBE_IS**	**RDA_IS**

Bits	Description	
[31:11]	**Reserved**	Reserved.
[10]	**ACON_ERR_IS**	**Auto Convention Error Interrupt Status Flag (Read Only)** This field indicates auto convention sequence error. If the received TS at ATR state is neither 0x3B nor 0x3F, this bit will be set. **Note:** This bit is read only, but it can be cleared by writing 1 to it.
[9]	**RTMR_IS**	**Receiver Buffer Time-out Interrupt Status Flag (Read Only)** This field is used for receiver buffer time-out interrupt status flag. **Note:** This field is the status flag of receiver buffer time-out state. If software wants to clear this bit, software must read all receiver buffer remaining data by reading SC_RBR buffer,
[8]	**INIT_IS**	**Initial End Interrupt Status Flag (Read Only)** This field is used for activation (ACT_EN(SC_ALTCTL[3])), deactivation (DACT_EN (SC_ALTCTL[2])) and warm reset (WARST_EN (SC_ALTCTL[4])) sequence interrupt status flag. **Note:** This bit is read only, but it can be cleared by writing 1 to it.
[7]	**CD_IS**	**Card Detect Interrupt Status Flag (Read Only)** This field is used for card detect interrupt status flag. The card detect status is CD_INS_F (SC_SR[12])] and CD_REM_F(SC_SR[11]). **Note:** This field is the status flag of CD_INS_F SC_SR[12]) SC_PINCSR[CD_INS_F] or CD_REM_F(SC_SR[11])]. So if software wants to clear this bit, software must write 1 to this field.
[6]	**BGT_IS**	**Block Guard Time Interrupt Status Flag (Read Only)** This field is used for block guard time interrupt status flag. **Note1:** This bit is valid when RX_BGT_EN (SC_ALTCTL[12]) is enabled.

		Note2: This bit is read only, but it can be cleared by writing “1” to it.
[5]	**TMR2_IS**	**Timer2 Interrupt Status Flag (Read Only)** This field is used for TMR2 interrupt status flag. **Note:** This bit is read only, but it can be cleared by writing 1 to it.
[4]	**TMR1_IS**	**Timer1 Interrupt Status Flag (Read Only)** This field is used for TMR1 interrupt status flag. **Note:** This bit is read only, but it can be cleared by writing 1 to it.
[3]	**TMR0_IS**	**Timer0 Interrupt Status Flag (Read Only)** This field is used for TMR0 interrupt status flag. **Note:** This bit is read only, but it can be cleared by writing 1 to it.
[2]	**TERR_IS**	**Transfer Error Interrupt Status Flag (Read Only)** This field is used for transfer error interrupt status flag. The transfer error states is at SC_SR register which includes receiver break error RX_EBR_F(SC_SR[6]), frame error RX_EFR_F(SC_SR[5], parity error RX_EPA_F(SC_SR[4] and receiver buffer overflow error RX_OVER_F(SC_SR[0]), transmit buffer overflow error TX_OVER_F(SC_SR[8]), receiver retry over limit error RX_OVER_REERR(SC_SR[22] and transmitter retry over limit error TX_OVER_REERR(SC_SC[30]). **Note:** This field is the status flag of RX_EBR_F(SC_SR[6]), RX_EFR_F(SC_SR[5], RX_EPA_F(SC_SR[4], RX_OVER_F(SC_SR[0]), TX_OVER_F(SC_SR[8]), RX_OVER_REERR(SC_SR[22] or TX_OVER_REERR(SC_SC[30]). So, if software wants to clear this bit, software must write 1 to each field.
[1]	**TBE_IS**	**Transmit Buffer Empty Interrupt Status Flag (Read Only)** This field is used for transmit buffer empty interrupt status flag. **Note:** This field is the status flag of transmit buffer empty state. If software wants to clear this bit, software must write data to SC_THR buffer and then this bit will be cleared automatically.
[0]	**RDA_IS**	**Receive Data Reach Interrupt Status Flag (Read Only)** This field is used for received data reaching trigger level RX_FTRI_LEV (SC_CTL[7:6]) interrupt status flag. **Note:** This field is the status flag of received data reaching RX_FTRI_LEV (SC_CTL[7:6]). If software reads data from SC_RBR and receiver buffer data byte number is less than RX_FTRI_LEV (SC_CTL[7:6]), this bit will be cleared automatically.

SC Transfer Status Register (SC_TRSR)

Register	Offset	R/W	Description	Reset Value
SC_TRSR	SCx_BA+0x20	R/W	SC Status Register.	0x0000_0202

31	30	29	28	27	26	25	24
TX_ATV	TX_OVER_REERR	TX_REERR	Reserved		TX_POINT_F		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RX_ATV	RX_OVER_REERR	RX_REERR	Reserved		RX_POINT_F		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		CD_PIN_F	CD_INS_F	CD_REM_F	TX_FULL_F	TX_EMPTY_F	TX_OVER_F
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	RX_EBR_F	RX_EFR_F	RX_EPA_F	Reserved	RX_FULL_F	RX_EMPTY_F	RX_OVER_F

Bits	Description	
[31]	**TX_ATV**	**Transmit In Active Status Flag (Read Only)** 0 = This bit is cleared automatically when TX transfer is finished or the last byte transmission has completed. 1 = This bit is set by hardware when TX transfer is in active and the STOP bit of the last byte has been transmitted.
[30]	**TX_OVER_REERR**	**Transmitter Over Retry Error (Read Only)** This bit is set by hardware when transmitter re-transmits over retry number limitation. **Note:** This bit is read only, but it can be cleared by writing 1 to it.
[29]	**TX_REERR**	**Transmitter Retry Error (Read Only)** This bit is set by hardware when transmitter re-transmits. **Note1:** This bit is read only, but it can be cleared by writing 1 to it. **Note2** This bit is a flag and cannot generate any interrupt to CPU.
[28:26]	**Reserved**	Reserved.
[25:24]	**TX_POINT_F**	**Transmit Buffer Pointer Status Flag (Read Only)** This field indicates the TX buffer pointer status flag. When CPU writes data into SC_THR, TX_POINT_F increases one. When one byte of TX Buffer is transferred to transmitter shift register, TX_POINT_F decreases one.
[23]	**RX_ATV**	**Receiver In Active Status Flag (Read Only)** This bit is set by hardware when RX transfer is in active. This bit is cleared automatically when RX transfer is finished.
[22]	**RX_OVER_REERR**	**Receiver Over Retry Error (Read Only)** This bit is set by hardware when RX transfer error retry over retry number limit. **Note1:** This bit is read only, but it can be cleared by writing 1 to it. **Note2:** If CPU enables receiver retries function by setting RX_ERETRY_EN (SC_CTL[19]), the RX_EPA_F(SC_SR[4]) flag will be ignored (hardware will not set RX_EPA_F(SC_SR[4])).

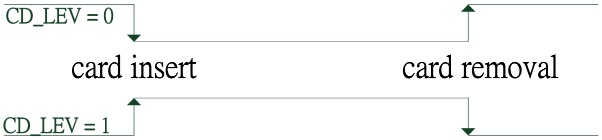
[21]	**RX_REERR**	**Receiver Retry Error (Read Only)** This bit is set by hardware when RX has any error and retries transfer. **Note1:** This bit is read only, but it can be cleared by writing 1 to it. **Note2** This bit is a flag and cannot generate any interrupt to CPU. **Note3:** If CPU enables receiver retry function by setting RX_ERETRY_EN (SC_CTL[19]) , the RX_EPA_F(SC_SR[4]) flag will be ignored (hardware will not set RX_EPA_F(SC_SR[4])).
[20:18]	**Reserved**	Reserved.
[17:16]	**RX_POINT_F**	**Receiver Buffer Pointer Status Flag (Read Only)** This field indicates the RX buffer pointer status flag. When SC receives one byte from external device, RX_POINT_F(SC_SR[17:16]) increases one. When one byte of RX buffer is read by CPU, RX_POINT_F(SC_SR[17:16]) decreases one.
[15:14]	**Reserved**	Reserved.
[13]	**CD_PIN_F**	**Card Detect Status Of SC_CD Pin Status (Read Only)** This bit is the pin status flag of SC_CD 0 = The SC_CD pin state at low. 1 = The SC_CD pin state at high.
[12]	**CD_INS_F**	**Card Detect Insert Status Of SC_CD Pin (Read Only)** This bit is set whenever card has been inserted. 0 = No effect. 1 = Card insert. **Note1:** This bit is read only, but it can be cleared by writing “1” to it. **Note2:** The card detect engine will start after SC_CEN (SC_CTL[0]) set.
[11]	**CD_REM_F**	**Card Detect Removal Status Of SC_CD Pin (Read Only)** This bit is set whenever card has been removal. 0 = No effect. 1 = Card removed. **Note1:** This bit is read only, but it can be cleared by writing “1” to it. **Note2:** Card detect engine will start after SC_CEN (SC_CTL[0])set.
[10]	**TX_FULL_F**	**Transmit Buffer Full Status Flag (Read Only)** This bit indicates TX buffer full or not.This bit is set when TX pointer is equal to 4, otherwise is cleared by hardware.
[9]	**TX_EMPTY_F**	**Transmit Buffer Empty Status Flag (Read Only)** This bit indicates TX buffer empty or not. When the last byte of TX buffer has been transferred to Transmitter Shift Register, hardware sets this bit high. It will be cleared when writing data into SC_THR (TX buffer not empty).
[8]	**TX_OVER_F**	**TX Overflow Error Interrupt Status Flag (Read Only)** If TX buffer is full, an additional write to SC_THR will cause this bit be set to “1” by hardware. **Note:** This bit is read only, but it can be cleared by writing 1 to it.
[7]	**Reserved**	Reserved.
[6]	**RX_EBR_F**	**Receiver Break Error Status Flag (Read Only)** This bit is set to logic 1 whenever the received data input (RX) held in the “spacing state” (logic 0) is longer than a full word transmission time (that is, the total time of “start bit” + data bits + parity + stop bits). . **Note1:** This bit is read only, but it can be cleared by writing 1 to it. **Note2:** If CPU sets receiver retries function by setting RX_ERETRY_EN(SC_CTL[19]) ,

		hardware will not set this flag.
[5]	**RX_EFR_F**	**Receiver Frame Error Status Flag (Read Only)** This bit is set to logic 1 whenever the received character does not have a valid “stop bit” (that is, the stop bit following the last data bit or parity bit is detected as a logic 0). **Note1:** This bit is read only, but it can be cleared by writing 1 to it. **Note2:** If CPU sets receiver retries function by setting RX_ERETRY_EN(SC_CTL[19]) , hardware will not set this flag.
[4]	**RX_EPA_F**	**Receiver Parity Error Status Flag (Read Only)** This bit is set to logic 1 whenever the received character does not have a valid “parity bit”. **Note1:** This bit is read only, but it can be cleared by writing 1 to it. **Note2:** If CPU sets receiver retries function by setting RX_ERETRY_EN(SC_CTL[19]) , hardware will not set this flag.
[3]	**Reserved**	Reserved.
[2]	**RX_FULL_F**	**Receiver Buffer Full Status Flag (Read Only)** This bit indicates RX buffer full or not. This bit is set when RX pointer is equal to 4, otherwise it is cleared by hardware.
[1]	**RX_EMPTY_F**	**Receiver Buffer Empty Status Flag(Read Only)** This bit indicates RX buffer empty or not. When the last byte of Rx buffer has been read by CPU, hardware sets this bit high. It will be cleared when SC receives any new data.
[0]	**RX_OVER_F**	**RX Overflow Error Status Flag (Read Only)** This bit is set when RX buffer overflow. If the number of received bytes is greater than Rx Buffer size (4 bytes), this bit will be set. **Note:** This bit is read only, but it can be cleared by writing 1 to it.

SC PIN Control State Register (SC_PINCSR)

Register	Offset	R/W	Description	Reset Value
SC_PINCSR	SCx_BA+0x24	R/W	SC Pin Control State Register.	0x0000_00x0

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**SC_DATA_I_ST**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**CD_LEV**	**SC_DATA_O**	**SC_OEN_ST**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADAC_CD_EN	**CLK_KEEP**	**Reserved**	**CD_PIN_ST**	**CD_INS_F**	**CD_REM_F**	**SC_RST**	**POW_EN**

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**SC_DATA_I_ST**	**SC Data Input Pin Status (Read Only)** This bit is the pin status of SC_DATA_I 0 = The SC_DATA_I pin is low. 1 = The SC_DATA_I pin is high.
[15:11]	**Reserved**	Reserved.
[10]	**CD_LEV**	**Card Detect Level** 0 = When hardware detects the card detect pin from high to low, it indicates a card is detected. 1 = When hardware detects the card detect pin from low to high, it indicates a card is detected.  **Note:** Software must select card detect level before Smart Card engine enable
[9]	**SC_DATA_O**	**SC Data Output Pin** This bit is the pin status of SC_DATA_O but user can drive SC_DATA_O pin to high or low by setting this bit. 0 = Drive SC_DATA_O pin to low. 1 = Drive SC_DATA_O pin to high. **Note:** When SC is at activation, warm reset or deactivation mode, this bit will be changed automatically. So don't fill this field when SC is in these modes.
[8]	**SC_OEN_ST**	**SC Data Output Enable Pin Status (Read Only)** This bit is the pin status of SC_DATA_OEN 0 = The SC_DATA_OEN pin state at low.

		1 = The SC_DATA_OEN pin state at high.
[7]	**ADAC_CD_EN**	**Auto Deactivation When Card Removal** 0 = Auto deactivation Disabled when hardware detected the card is removal. 1 = Auto deactivation Enabled when hardware detected the card is removal. **Note:** When the card is removal, hardware will stop any process and then do deactivation sequence (if this bit be setting). If this process completes. Hardware will generate an interrupt INT_INIT to CPU.
[6]	**CLK_KEEP**	**SC Clock Enable Control** 0 = SC clock generation Disabled. 1 = SC clock always keeps free running. **Note:** When operating in activation, warm reset or deactivation mode, this bit will be changed automatically. So don't fill this field when operating in these modes.
[5]	**Reserved**	Reserved.
[4]	**CD_PIN_ST**	**Card Detect Status Of SC_CD Pin Status (Read Only)** This bit is the pin status flag of SC_CD 0 = SC_CD pin state at low. 1 = SC_CD pin state at high.
[3]	**CD_INS_F**	**Card Detect Insert Status Of SC_CD Pin (Read Only)** This bit is set whenever card has been inserted. 0 = No effect. 1 = Card insert. **Note1:** This bit is read only, but it can be cleared by writing "1" to it. **Note2:** Card detect engine will start after SC_CTL [SC_CEN] set.
[2]	**CD_REM_F**	**Card Detect Removal Status Of SC_CD Pin (Read Only)** This bit is set whenever card has been removal. 0 = No effect. 1 = Card Removal. **Note1:** This bit is read only, but it can be cleared by writing "1" to it. **Note2:** Card detect engine will start after SC_CTL [SC_CEN] set.
[1]	**SC_RST**	**SC_RST Pin Signal** This bit is the pin status of SC_RST but user can drive SC_RST pin to high or low by setting this bit. Write this field to drive SC_RST pin. 0 = Drive SC_RST pin to low. 1 = Drive SC_RST pin to high. Read this field to get SC_RST pin status. 0 = SC_RST pin status is low. 1 = SC_RST pin status is high. **Note:** When operating at activation, warm reset or deactivation mode, this bit will be changed automatically. So don't fill this field when operating in these modes.
[0]	**POW_EN**	**SC_POW_EN Pin Signal** Software can set POW_EN and POW_INV to decide SC_PWR pin is in high or low level. Write this field to drive SC_PWR pin Refer POW_INV description for programming SC_PWR pin voltage level. Read this field to get SC_PWR pin status. 0 = SC_PWR pin status is low. 1 = SC_PWR pin status is high. **Note:** When operating at activation, warm reset or deactivation mode, this bit will be

		changed automatically. So don’t fill this field when operating in these modes.

SC Timer Control Register 0 (SC_TMR0)

Register	Offset	R/W	Description	Reset Value
SC_TMR0	SCx_BA+0x28	R/W	SC Internal Timer Control Register 0.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				MODE			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CNT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**MODE**	**Timer 0 Operation Mode Selection** This field indicates the internal 24-bit timer operation selection.
[23:0]	**CNT**	**Timer 0 Counter Value (ETU Base)** This field indicates the internal timer operation values.

SC Timer Control Register 1 (SC_TMR1)

Register	Offset	R/W	Description	Reset Value
SC_TMR1	SCx_BA+0x2C	R/W	SC Internal Timer Control Register 1.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				MODE			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CNT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**MODE**	**Timer 1 Operation Mode Selection** This field indicates the internal 8-bit timer operation selection.
[7:0]	**CNT**	**Timer 1 Counter Value (ETU Base)** This field indicates the internal timer operation values.

SC Timer Control Register 2 (SC_TMR2)

Register	Offset	R/W	Description	Reset Value
SC_TMR2	SCx_BA+0x30	R/W	SC Internal Timer Control Register 2.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				**MODE**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CNT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT							

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:24]	**MODE**	**Timer 2 Operation Mode Selection** This field indicates the internal 8-bit timer operation selection
[7:0]	**CNT**	**Timer 2 Counter Value (ETU Base)** This field indicates the internal timer operation values.

SC UART Mode Control Register (SCx_UACTL)

Register	Offset	R/W	Description	Reset Value
SC_UACTL	SCx_BA + 0x34	R/W	SC UART Mode Control Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OPE	PBDIS	DATA_LEN		Reserved			UA_MODE_EN

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**OPE**	**Odd Parity Enable Control** 0 = Even number of logic 1's are transmitted or check the data word and parity bits in receiving mode. 1 = Odd number of logic 1's are transmitted or check the data word and parity bits in receiving mode. **Note:** This bit has effect only when PBDIS bit is '0'.
[6]	**PBDIS**	**Parity Bit Disable Control** 0 = Parity bit is generated or checked between the "last data word bit" and "stop bit" of the serial data. 1 = Parity bit is not generated (transmitting data) or checked (receiving data) during transfer. **Note:** In smart card mode, this field must be '0' (default setting is with parity bit)
[5:4]	**DATA_LEN**	**Data Length** **00 = Character Data Length is 8 bits.** **01 = Character Data Length is 7 bits.** **10 = Character Data length is 6 bits.** **11 = Character Data Length is 5 bits.** **Note:** In smart card mode, this DATA_LEN must be '00'
[3:1]	**Reserved**	Reserved.
[0]	**UA_MODE_EN**	**UART Mode Enable Control** 0 = Smart Card mode. 1 = UART mode. **Note1:** When operating in UART mode, user must set CON_SEL (SC_CTL[5:4]) = 00 and AUTO_CON_EN(SC_CTL[3]) = 0. **Note2:** When operating in Smart Card mode, user must set SC_UACTL [7:0] = 00. **Note3:** When UART is enabled, hardware will generate a reset to resetFIFO and internal state machine.

SC Timer Current Data Register A (SC_TDRA)

Register	Offset	R/W	Description	Reset Value
SC_TDRA	SCx_BA+0x38	R	SC Timer Current Data Register A.	0x0000_07FF

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TDR0							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TDR0							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TDR0							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**TDR0**	**Timer0 Current Data Value (Read Only)** This field indicates the current count values of timer0.

SC Timer Current Data Register B (SC_TDRB)

Register	Offset	R/W	Description	Reset Value
SC_TDRB	SCx_BA+0x3C	R	SC Timer Current Data Register B.	0x0000_7F7F

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TDR2							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TDR1							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**TDR2**	**Timer2 Current Data Value (Read Only)** This field indicates the current count values of timer2.
[7:0]	**TDR1**	**Timer1 Current Data Value (Read Only)** This field indicates the current count values of timer1.

6.16 I^{2}C

6.16.1 Overview

I^{2}C is a two-wire, bi-directional serial bus that provides a simple and efficient method of data exchange between devices. The I^{2}C standard is a true multi-master bus including collision detection and arbitration that prevents data corruption if two or more masters attempt to control the bus simultaneously. Serial, 8-bit oriented bi-directional data transfers can be made up to 1 Mbps.

Data is transferred between a Master and a Slave synchronously to SCL on the SDA line on a byte-by-byte basis. Each data byte is 8-bit long. There is one SCL clock pulse for each data bit with the MSB being transmitted first. An acknowledge bit follows each transferred byte.

A transition on the SDA line while SCL is high is interpreted as a command (START or STOP). Each bit is sampled during the high period of SCL; therefore, the SDA line may be changed only during the low period of SCL and must be held stable during the high period of SCL.

The controller's on-chip I^{2}C logic provides the serial interface that meets the I^{2}C bus standard mode specification. The I^{2}C controller handles byte transfers autonomously. Pull up resistor is needed for I^{2}C operation as these are open drain pins.

The I^{2}C controller is equipped with two slave address registers. The contents of the registers are irrelevant when I^{2}C is in Master mode. In the Slave mode, the seven most significant bits must be loaded with the user's own slave address. The I^{2}C hardware will react if the contents of I2CADDR are matched with the received slave address.

This controller supports the "General Call (GC)" function. If the GCALL (I2CSADDR[0]) bit is set this controller will respond to General Call address (00H). Clear GC bit to disable general call function. When GCALL bit is set and the I^{2}C is in Slave mode, it can receive the general call address which is equal to 00H after master sends general call address to the I^{2}C bus, then it will follow status of GC mode. If it is in Master mode, the ACK bit must be cleared when it sends general call address of 00H to the I^{2}C bus.

The I^{2}C-bus controller supports multiple address recognition with two address mask register. When the bit in the address mask register is set to one, it means the received corresponding address bit is don't-care. If the bit is set to 0, that means the received corresponding register bit should be exact the same as address register.

6.16.2 Features

- Supports two I^{2}C channels and both of them can acts as Master or Slave mode
- Bidirectional data transfer between masters and slaves
- Multi-master bus (no central master)
- Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
- Serial clock synchronization allows devices with different bit rates to communicate via one serial bus
- Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer
- One built-in 14-bit time-out counter requesting the I^{2}C interrupt if the I^{2}C bus hangs up and timer-out counter overflows.
- Programmable clock divider allows versatile rate control
- Supports 7-bit addressing mode

- Supports multiple address recognition (Two slave addresses with mask option)
- Supports Power-down wake-up function
- Supports two-Level FIFO

6.16.3 Basic Configuration

The basic configurations of I^2C are as follows:

- I2C0 pin configuration: PA13, PA12 (GPA_H_MFP[23:16]) / PB3, PB2 (GPB_L_MFP[15:8]) / PC1, PC0 (GPC_L_MFP[7:0])
- I2C1 pin configuration: PA15, PA14 (GPA_H_MFP[31:24]) / PC3, PC2 (GPC_L_MFP[15:8]) / PC11, PC10 (GPC_H_MFP[15:8]) and PA5 GPA_L_MFP[23:20]
- Enable I2C clock (I2C_EN) on APBCLK [9:8] register.
- Reset I2C controller (I2C_RST) on IPRSTC2 [9:8] register.

6.16.4 Functional Description

On I^2C bus, data is transferred between a Master and a Slave. Data bits transfer on the SCL and SDA lines are synchronously on a byte-by-byte basis. Each data byte is 8-bit long. There is one SCL clock pulse for each data bit with the MSB being transmitted first, and an acknowledge bit follows each transferred byte. Each bit is sampled during the high period of SCL; therefore, the SDA line may be changed only during the low period of SCL and must be held stable during the high period of SCL. A transition on the SDA line while SCL is high is interpreted as a command (START or STOP). Please refer to the following figure for more detailed I^2C BUS Timing.

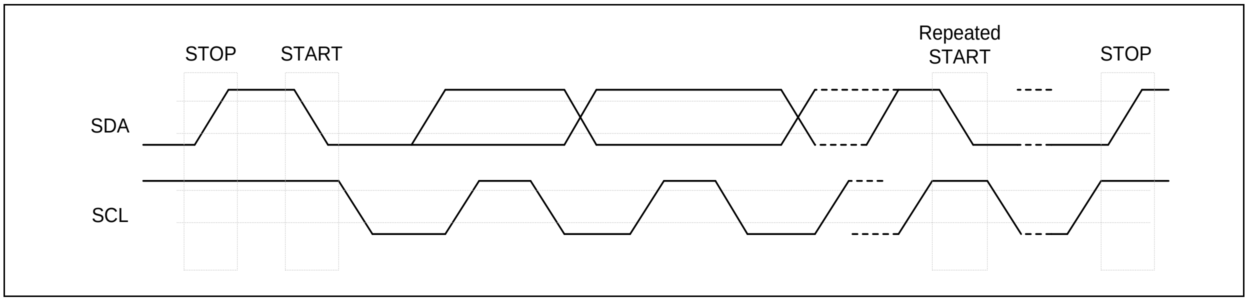


Figure 6-74 I^2C Bus Timing

The device's on-chip I^2C provides the serial interface that meets the I^2C bus standard mode specification. The I^2C port handles byte transfers autonomously. To enable this port, the bit IPEN (I2CON[0]) should be s't''o''1'. The I^2C hardware interfaces to the I^2C bus via two pins: SDA and SCL. When I/O pins are used as I^2C ports, user must set the pins function to I^2C in advance.

There is two-level FIFO to improve the performance of I^2C bus. In two-level FIFO mode, the next transmitted or the last received data can be active even if the current data is transmitted or the last received isn't read back yet.

The I^2C SCL bus is stretched low when there is SI event. The NOSTRETCH (I2CON2[5]) bit is used to force the I^{2}C SCL bus is no stretched under the SI event.

There are under run or over run interrupt when the two-level FIFO mode (TWOFF_EN (I2CON2[4]) is enabled and the interrupt event UNDER_INTEN (I2CON2[2]) and OVER_INTEN (I2CON2[1]) enable is set.

Note: Pull-up resistor is needed for I^{2}C operation as the SDA and SCL are open-drain pins.

6.16.4.1 I2C Protocol

The following figure shows the typical I^{2}C protocol. Normally, a standard communication consists of four parts:

- START or Repeated START signal generation
- STOP signal generation
- Slave address transfer
- Data transfer

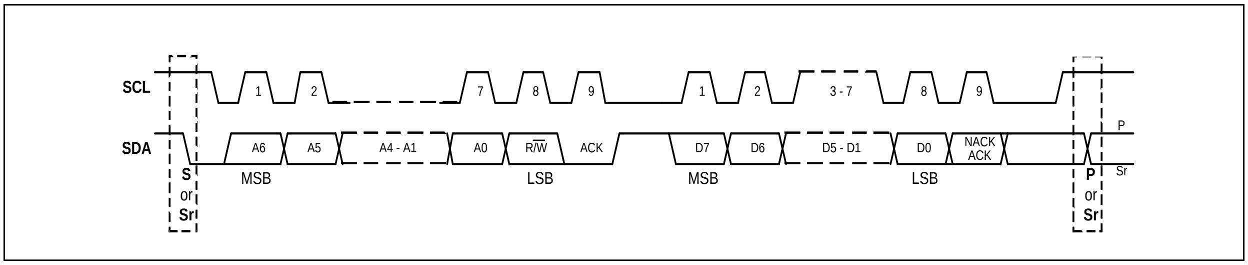


Figure 6-75 I^{2}C Protocol

6.16.4.2 START or Repeated START signal

When the bus is free/idle, meaning no master device is engaging the bus (both SCL and SDA lines are high), a master can initiate a transfer by sending a START signal. A START signal, usually referred to as the S-bit, is defined as a HIGH to LOW transition on the SDA line while SCL is HIGH. The START signal denotes the beginning of a new data transfer.

A Repeated START (Sr) is a START signal without first generating a STOP signal. The master uses this method to communicate with another slave or the same slave in a different transfer direction (e.g. from writing to a device to reading from a device) without releasing the bus.

6.16.4.3 STOP signal

The master can terminate the communication by generating a STOP signal. A STOP signal, usually referred to as the STOP-bit, is defined as a LOW to HIGH transition on the SDA line while SCL is HIGH.

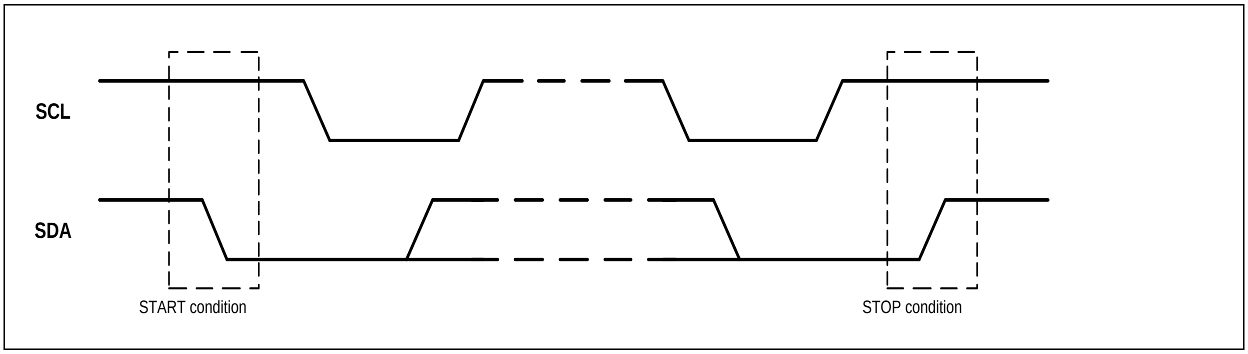


Figure 6-76 I^{2}C START and STOP Conditions

6.16.4.4 Slave Address Transfer

The first byte of data transferred by the master immediately after the START signal is the slave address. This is a 7-bits calling address followed by the R/W bit. The R/W bit signals the slave the data transfer direction. No two slaves in the system can have the same address. Only the slave with an address that matches the one transmitted by the master will respond by returning an acknowledge bit by pulling the SDA low at the 9th SCL clock cycle.

6.16.4.5 Data Transfer

When a slave receives a correct address with an R/W bit, the data will follow R/W bit specified to transfer. Each transferred byte is followed by an acknowledge bit on the 9th SCL clock cycle. If the slave signals a Not Acknowledge (NACK), the master can generate a STOP signal to abort the data transfer or generate a Repeated START signal and start a new transfer cycle.

If the master, as the receiving device, does Not Acknowledge (NACK) the slave, the slave releases the SDA line for the master to generate a STOP or Repeated START signal.

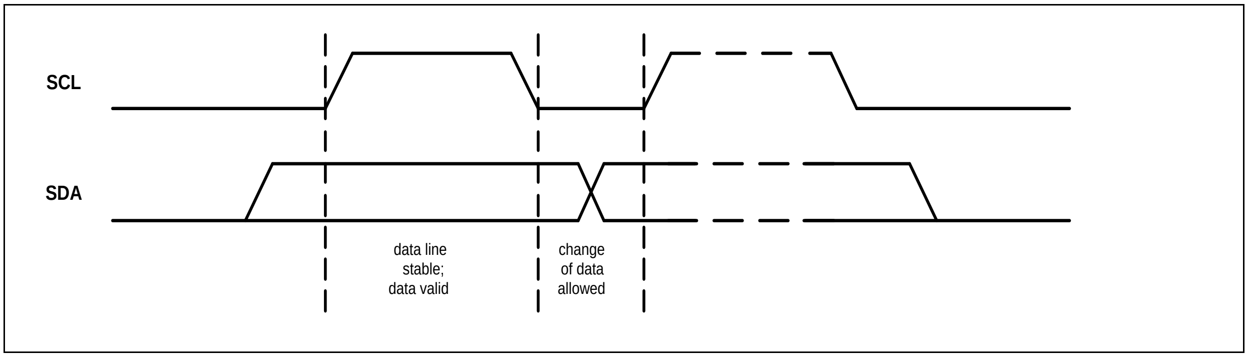


Figure 6-77 Bit Transfer on I^{2}C Bus

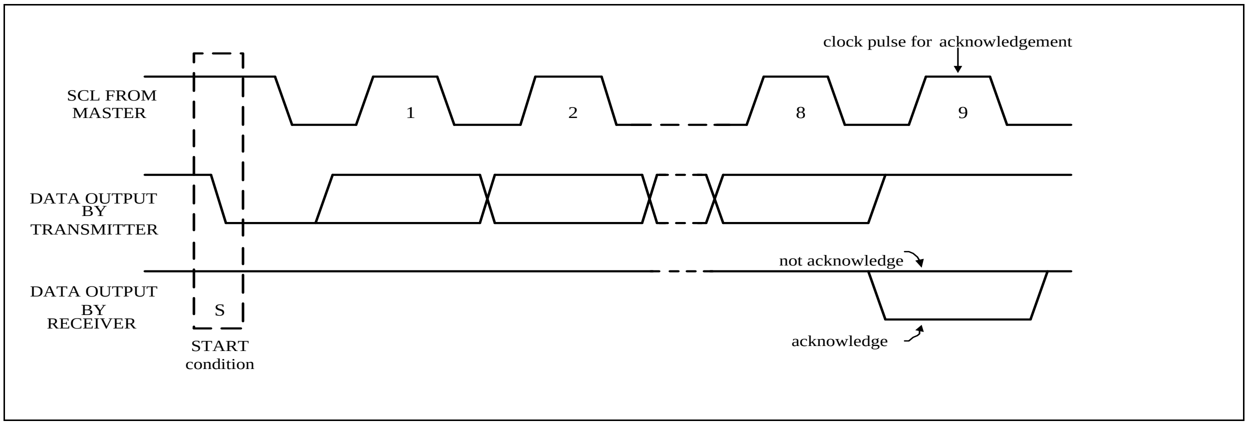


Figure 6-78 Acknowledge on I^{2}C Bus

6.16.4.6 I2C Protocol Register

To control I^{2}C port through the following special function registers: I2CON (control register), I2CINTSTS (interrupt status register), I2CSTATUS (status register), I2DIV (clock rate register), I2CTOUT (time-out counter register), I2CDAT (data register), I2CSADDRn (address registers, n=0~1), I2CSAMASKn (address mask registers, n=0~1), I2CON2(control register 2) and I2CSTATUS2 (status register2).

6.16.4.7 Control Register (I2CON)

The CPU can read from and write to I2CON[7:0] directly. When the I^{2}C port is enabled by setting IPEN (I2CON [0]) to high, the internal states will be controlled by I2CON and I^{2}C logic hardware.

There are two bits are affected by hardware: the INTSTS (I2CINTSTS[0]) bit is set when the I^{2}C hardware requests a serial interrupt, and the STOP (I2CON[2]) bit is cleared when a STOP condition is present on the bus. The STOP bit is also cleared when IPEN = 0.

Once a new status code is generated and stored in I2CSTATUS, the I^{2}C Interrupt Flag bit INTSTS (I2CINTSTS [0]) will be set automatically. If the Enable Interrupt bit INTEN (I2CON [7]) is set at this time, the I^{2}C interrupt will be generated. The bit field I2CSTATUS[7:0] stores the internal state code, the content keeps stable until INTSTS is cleared by software.

6.16.4.8 Interrupt Status Register (I2CINTSTS)

There are 3 interrupt status.

(1). INTSTS: When a new state is present in the I2CSTATUS register, this bit will be set automatically, and if INTEN bit is set, the I^{2}C interrupt is requested.

(2). TIF: Refer to the section I^{2}C time out counter.

(3). WAKEUP_ACK_DONE: The ACK bit cycle of address match frame is done in power-down. The controller will stretch the SCL to low when the address is matched the device's slave address and the ACK cycle done. The SCL is stretched until the bit is clear by user. If the frequency of SCL is low speed and the system has wake-up from address match frame, the user shall check this bit to confirm this frame has transaction done and then to do the wake-up procedure.

6.16.4.9 Status Register (I2CSTATUS)

I2CSTATUS[7:0] is an 8-bit read-only register. The bit field I2CSTATUS[7:0] contain the status code. There are 26 possible status codes. When I2CSTATUS[7:0] is F8H, no serial interrupt is requested. All other I2CSTATUS[7:0] values correspond to defined I^{2}C states. When each of these states is entered, a status interrupt is requested (INTSTS (I2CINTSTS[0]) = 1). A valid status code is present in I2CSTATUS[7:0] one cycle after INTSTS is set by hardware and is still present one cycle after INTSTS has been reset by software.

In addition, the state 00H stands for a Bus Error, which occurs when a START or STOP condition is present at an incorrect position in the I^{2}C format frame. A Bus Error may occur during the serial transfer of an address byte, a data byte or an acknowledge bit. To recover I^{2}C from bus error, STOP (I2CON[2]) should be set and INTSTS (I2CINTSTS[0]) should be clear to enter Not Addressed Slave mode. Then STOP is cleared to release bus and to wait for a new communication. I^{2}C bus cannot recognize stop condition during this action when bus error occurs.

Master Mode		**Slave Mode**	
STATUS	**Description**	**STATUS**	**Description**
0x08	Start	0xA0	Slave Transmit Repeat Start or Stop
0x10	Master Repeat Start	0xA8	Slave Transmit Address ACK
0x18	Master Transmit Address ACK	0xB0	Slave Transmit Arbitration Lost
0x20	Master Transmit Address NACK	0xB8	Slave Transmit Data ACK
0x28	Master Transmit Data ACK	0xC0	Slave Transmit Data NACK
0x30	Master Transmit Data NACK	0xC8	Slave Transmit Last Data ACK
0x38	Master Arbitration Lost	0x60	Slave Receive Address ACK
0x40	Master Receive Address ACK	0x68	Slave Receive Arbitration Lost
0x48	Master Receive Address NACK	0x80	Slave Receive Data ACK
0x50	Master Receive Data ACK	0x88	Slave Receive Data NACK
0x58	Master Receive Data NACK	0x70	GC mode Address ACK
0x00	Bus error	0x78	GC mode Arbitration Lost
		0x90	GC mode Data ACK
		0x98	GC mode Data NACK
0xF8	Bus Released **Note:** Status “0xF8” exists in both master/slave modes, and it won’t raise interrupt.		

Table 6-20 I^{2}C Status Code Description

6.16.4.10 I^{2}C Baud Rate Bits (I2CDIV)

The data baud rate of I^{2}C is determined by CLK_DIV(I2CDIV[7:0]) register when I^{2}C is in Master mode. It is not necessary in a Slave mode. In Slave mode, I^{2}C will automatically synchronize with any clock frequency from master I^{2}C device.

The data baud rate of I^{2}C setting is Data Baud Rate of I2C = (system clock) / (4x (CLK_DIV +1)). If system clock = 16 MHz, the CLK_DIV = 40 (28H), the data baud rate of I2C = 16 MHz / (4x (40 +1)) = 97.5K bits/sec.

6.16.4.11 The I^{2}C Time-out Counter (I2CTOUT)

There is a 14-bits time-out counter which can be used to deal with the I^{2}C bus hang-up. If the time-out counter is enabled, the counter starts up counting until it overflows (TIF (I2CINTSTS[1])=1) and generates I^{2}C interrupt to CPU or stops counting by clearing INTEN (I2CON[7]) to 0. When time-out counter is enabled, setting flag INTSTS (I2CINTSTS[0]) to high will reset the counter and re-start counting after INTSTS is cleared. If I^{2}C bus is hung up, it causes the I2CSTATUS and flag INTSTS are not updated for a period, the 14-bit time-out counter may overflow and acknowledge CPU the I^{2}C interrupt. Refer to the following figure for the 14-bit time-out counter. User may write 1 to clear TIF to 0.

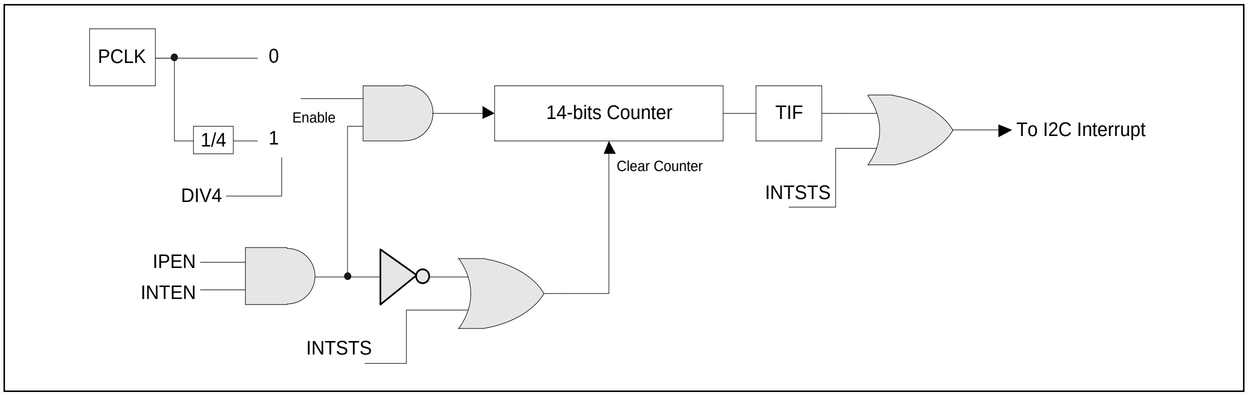


Figure 6-79 I^{2}C Time-out Block Diagram

6.16.4.12 I^{2}C Data Register (I2CDATA)

This register contains a byte of serial data to be transmitted or a byte which just has been received. The CPU can read from or write to this 8-bit (DATA(I2CDATA[7:0])) directly while it is not in the process of shifting a byte. When I^{2}C is in a defined state and the serial interrupt flag INTSTS (I2CINTSTS[0]) is set, data in DATA remains stable. While data is being shifted out, data on the bus is simultaneously being shifted in; DATA always contains the last data byte present on the bus.

The acknowledge bit is controlled by the I^{2}C hardware and cannot be accessed by the CPU. Serial data is shifted through into DATA on the rising edges of serial clock pulses on the SCL line. When a byte has been shifted into DATA, the serial data is available in DATA, and the acknowledge bit (ACK or NACK) is returned by the control logic during the ninth clock pulse. In order to monitor bus status while sending data, the bus data will be shifted to DATA[7:0] when sending DATA[7:0] to bus. In the case of sending data, serial data bits are shifted out from I2CDAT [7:0] on the falling edge of SCL clocks, and is shifted to DATA [7:0] on the rising edge of SCL clocks.

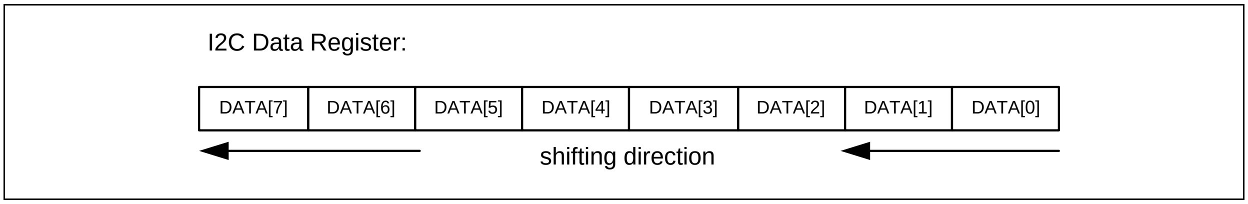


Figure 6-80 I^{2}C Data Shifting Direction

6.16.4.13 Address Registers (I2CSADDR)

The I^{2}C port is equipped with four slave address registers I2CSADDRn (n=0~1). The contents of the register are irrelevant when I^{2}C is in Master mode. In Slave mode, the bit field I2CSADDRn[7:1] must be loaded with the chip's own slave address. The I^{2}C hardware will react if the contents of I2CSADDRn are matched with the received slave address.

The I^{2}C ports support the "General Call" function. If the GCALL (I2CSADDRn[0]) bit is set, the I^{2}C port hardware will respond to General Call address (00H). Clearing GCALL bit will disable general call function.

When GCALL bit is set and the I^{2}C is in Slave mode, it can receive the general call address by 00H after Master send general call address to I^{2}C bus, then it will follow status of GCALL mode.

6.16.4.14 Slave Address Mask Registers (I2CSAMASK)

I^{2}C bus controllers support multiple address recognition with two address mask registers. When the bit in the address mask register is set to 1, it means the received corresponding address bit is don't-care. If the bit is set to 0, that means the received corresponding register bit should be exact the same as address register.

6.16.4.15 I^{2}C Control Register 2 (I2CON2)

The bus clock is stretched automatically when the INTSTS (I2CINTSTS[0]) is set. The NOSTRETCH (I2CON2[5]) bit is used to no stretch the bus clock when this bit is set to 1.

For the TWOFF_EN (I2CON2[4]) bit , it is used to enable the two-level FIFO for I^{2}C transmitted or received buffer. It is used to improve the performance of the I^{2}C bus. If this bit is set = 1, the control bit of START (I2CON[3]) for repeat start or STOP (I2CON[2]) bit should be set after the current INTSTS is clear. For example: if there are 4 data shall be transmitted and then stop it. The STOP bit shall be set after the 3rd data's INTSTS event being clear. In this time, the 4th data can be transmitted and the I^{2}C stop after the 4th data transmission done.

The two level FIFO status interrupt enable can be enabled by I2CON2[3:2] to generate the underrun or overrun event.

When enters Power-down mode, other I^{2}C master can wake up our chip by addressing our I^{2}C device, user must configure the related setting WKUP_EN (I2CON2[0]) before entering Power-down mode. When the system is wake-up by the matched address frame, the user shall check the WAKEUP_ACK_DONE (I2CINTSTS[7]) bit is set to 1 to confirm the address wake-up frame has done. Then, the user can read the WR_STATUS (I2CSTATUS2[3]) to know the bus transaction direction in the next frame. The bus of SCL is stretched after the ACK bit cycle done. It can't be released before clearing the WAKEUP_ACK_DONE bit to 0.

6.16.4.16 I^{2}C Status Register 2 (I2CSTATUS2)

The two level FIFO status, busy free information, FIFO empty, FIFO full and overrun or underun are also list in the this register.

The WR_STATUS (I2CSTATUS2[3]) bit records the Read/Write command on the address match wake-up frame. The user can use read this bit's status to prepare the next transmitted data (WR_STATUS = 0) or to wait the incoming data (WR_STATUS = 1) can be stored in time after the system is wake-up by the address match frame.

When system is woken up by other I^{2}C master device, WKUPIF (I2CSTATUS2[0]) is set to indicate this event. User needs write "1" to clear this bit. The other status bits are used to indicate the current FIFO status when the TWOFF_EN (I2CON2[4]) is set.

6.16.4.17 Operation Mode

The on-chip I^{2}C ports support three operation modes, Master, Slave, and General Call Mode.

In a given application, I2C port may operate as a master or as a slave. In Slave mode, the I2C port hardware looks for its own slave address and the general call address. If one of these addresses is detected, and if the slave is willing to receive or transmit data from/to master(by setting the ACK (I2CON[1]) bit), acknowledge pulse will be transmitted out on the 9th clock, hence an interrupt is requested on both master and slave devices if interrupt is enabled. When the microcontroller wishes to become the bus master, hardware waits until the bus is free before entering Master mode so that a possible slave action is not be interrupted. If bus arbitration is lost in Master mode, I2C port switches to Slave mode immediately and can detect its own slave address in the same serial transfer.

To control the I2C bus transfer in each mode, user needs to set I2CON, I2CDATA registers according to current status code of I2CSTATUS register. In other words, for each I2C bus action, user needs to check current status by I2CSTATUS register, and then set I2CON, I2CDATA registers to take bus action. Finally, check the response status by I2CSTATUS.

The bits, START, STOP and ACK (I2CON[3:1]) are used to control the next state of the I2C hardware after INTSTS flag of I2CINTSTS [0] register is cleared. Upon completion of the new action, a new status code will be updated in I2CSTATUS register and the INTSTS flag (I2CINTSTS[0]) will be set. If the I2C interrupt control bit INTEN (I2CON [7]) is set, appropriate action or software branch of the new status code can be performed in the Interrupt service routine.

The following figure shows the current I2C status code is 0x08, and then set DATA=SLA+W and (STA,STO,SI,AA) = (0,0,1,x) to send the address to I2C bus. If a slave on the bus matches the address and response ACK, the I2CSTATUS will be updated by status code 0x18.

Note: (STA, STO, SI, AA) = (START, STOP, INTSTS, ACK)

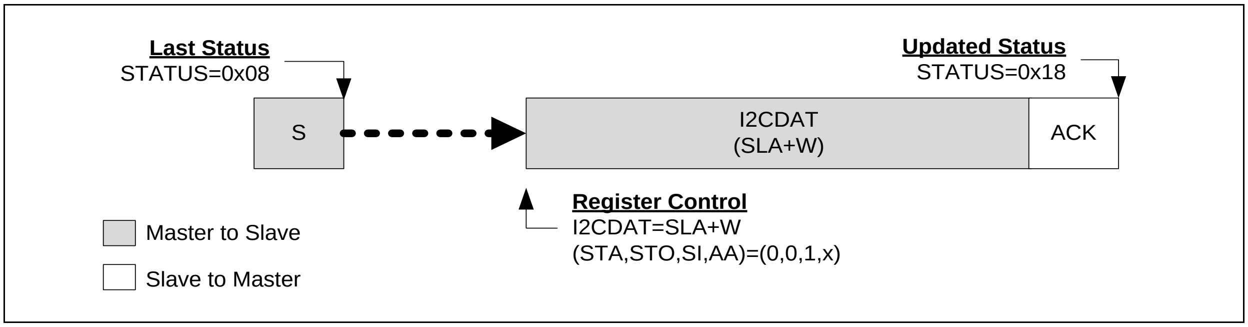


Figure 6-81 Control I2C Bus according to Current I2C Status

6.16.4.18 Master Mode

In the following figures, all possible protocols for I2C master are shown. User needs to follow proper path of the flow to implement required I2C protocol.

In other words, user can send a START signal to bus and I2C will be in Master Transmitter mode or Master receiver mode after START signal has been sent successfully and new status code would be 0x08. Followed by START signal, user can send slave address, read/write bit, data and Repeat START, STOP to perform I2C protocol.

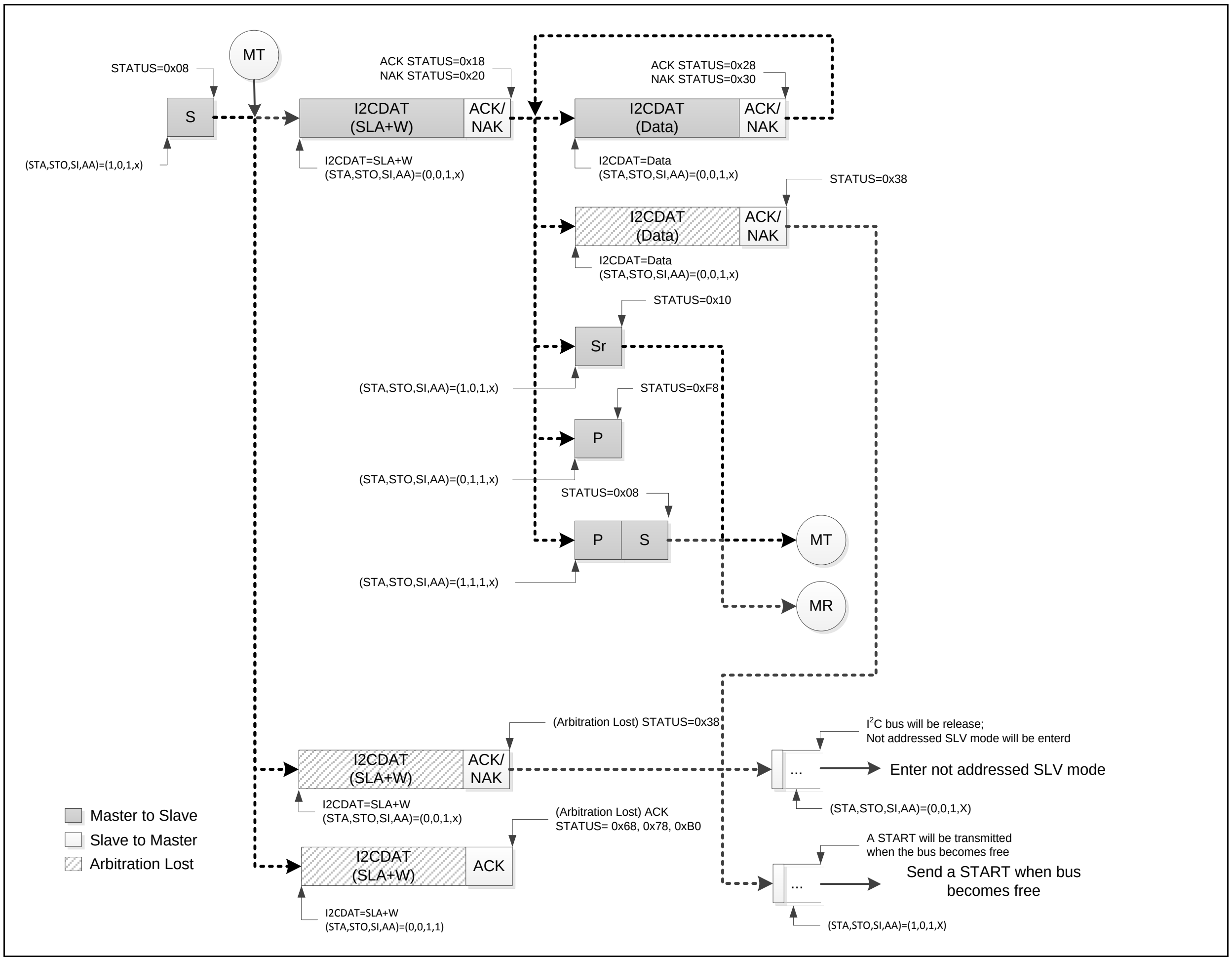


Figure 6-82 Master Transmitter Mode Control Flow

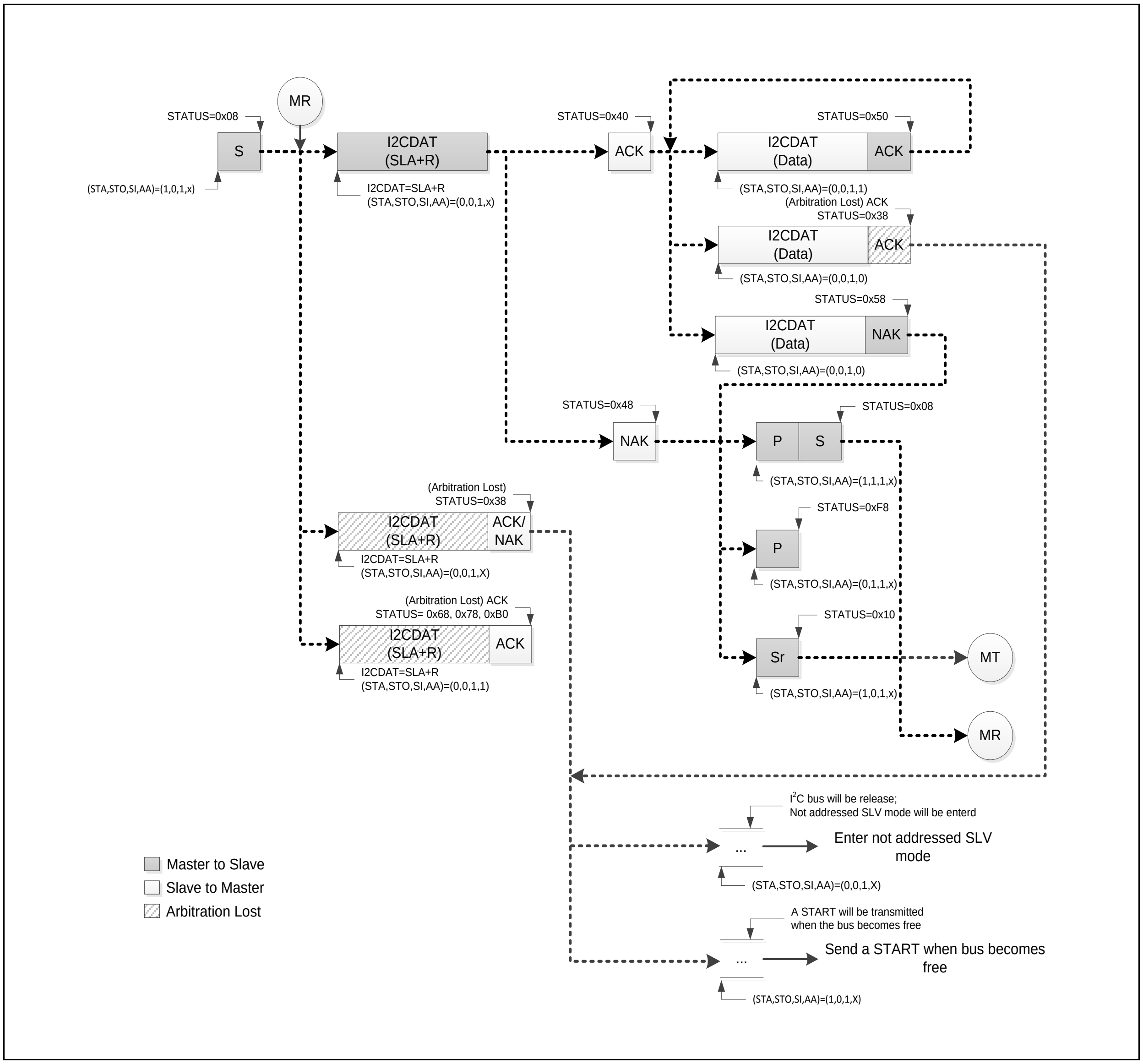

MR
STATUS=0x08
S
(STA,STO,SI,AA)=(1,0,1,x)
I2CDAT
(SLA+R)
I2CDAT=SLA+R
(STA,STO,SI,AA)=(0,0,1,x)
STATUS=0x40
ACK
STATUS=0x50
I2CDAT
(Data)
ACK
(STA,STO,SI,AA)=(0,0,1,1)
(Arbitration Lost) ACK
STATUS=0x38
I2CDAT
(Data)
ACK
(STA,STO,SI,AA)=(0,0,1,0)
STATUS=0x58
I2CDAT
(Data)
NAK
(STA,STO,SI,AA)=(0,0,1,0)
STATUS=0x48
NAK
STATUS=0x08
P
S
(STA,STO,SI,AA)=(1,1,1,x)
(Arbitration Lost)
STATUS=0x38
I2CDAT
(SLA+R)
ACK/
NAK
I2CDAT=SLA+R
(STA,STO,SI,AA)=(0,0,1,X)
STATUS=0xF8
P
(STA,STO,SI,AA)=(0,1,1,x)
(Arbitration Lost) ACK
STATUS= 0x68, 0x78, 0xB0
I2CDAT
(SLA+R)
ACK
I2CDAT=SLA+R
(STA,STO,SI,AA)=(0,0,1,1)
STATUS=0x10
Sr
MT
(STA,STO,SI,AA)=(1,0,1,x)
MR
I²C bus will be release;
Not addressed SLV mode will be enterd
...
Enter not addressed SLV mode
(STA,STO,SI,AA)=(0,0,1,X)
A START will be transmitted
when the bus becomes free
...
Send a START when bus becomes free
(STA,STO,SI,AA)=(1,0,1,X)
Master to Slave
Slave to Master
Arbitration Lost

Figure 6-83 Master Receiver Mode Control Flow

If the I^{2}C is in Master mode and gets arbitration lost, the status code will be 0x38. In status 0x38, user may set (START, STOP, INTSTS, ACK) = (1, 0, 1, X) to send START to re-start Master operation when bus become free. Otherwise, user may set (START, STOP, INTSTS, ACK) = (0, 0, 1, X) to release I^{2}C bus and enter not addressed Slave mode.

Note: (STA, STO, SI, AA) = (START, STOP, INTSTS, ACK)

6.16.4.19 Slave Mode

When reset default, I^{2}C is not addressed and will not recognize the address on I^{2}C bus. User can set slave address by I2CSADDRx and set (START, STOP, INTSTS, ACK) = (0, 0, 1, 1) to let I^{2}C recognize the address sent by master. The follow figure shows all the possible flow for I^{2}C in Slave mode. Users need to follow a proper flow to implement their own I^{2}C protocol.

If bus arbitration is lost in Master mode, I^{2}C port switches to Slave mode immediately and can detect its own slave address in the same serial transfer. If the detected address is SLA+W (Master want to write data to Slave) after arbitration lost, the status code is 0x68. If the detected address is SLA+R (Master want to read data from Slave) after arbitration lost, the status code is 0xB0.

Note: During I^{2}C communication, the SCL clock will be released when writing ‘1’ to clear INTSTS flag in Slave mode.

Note: (STA, STO, SI, AA) = (START, STOP, INTSTS, ACK)

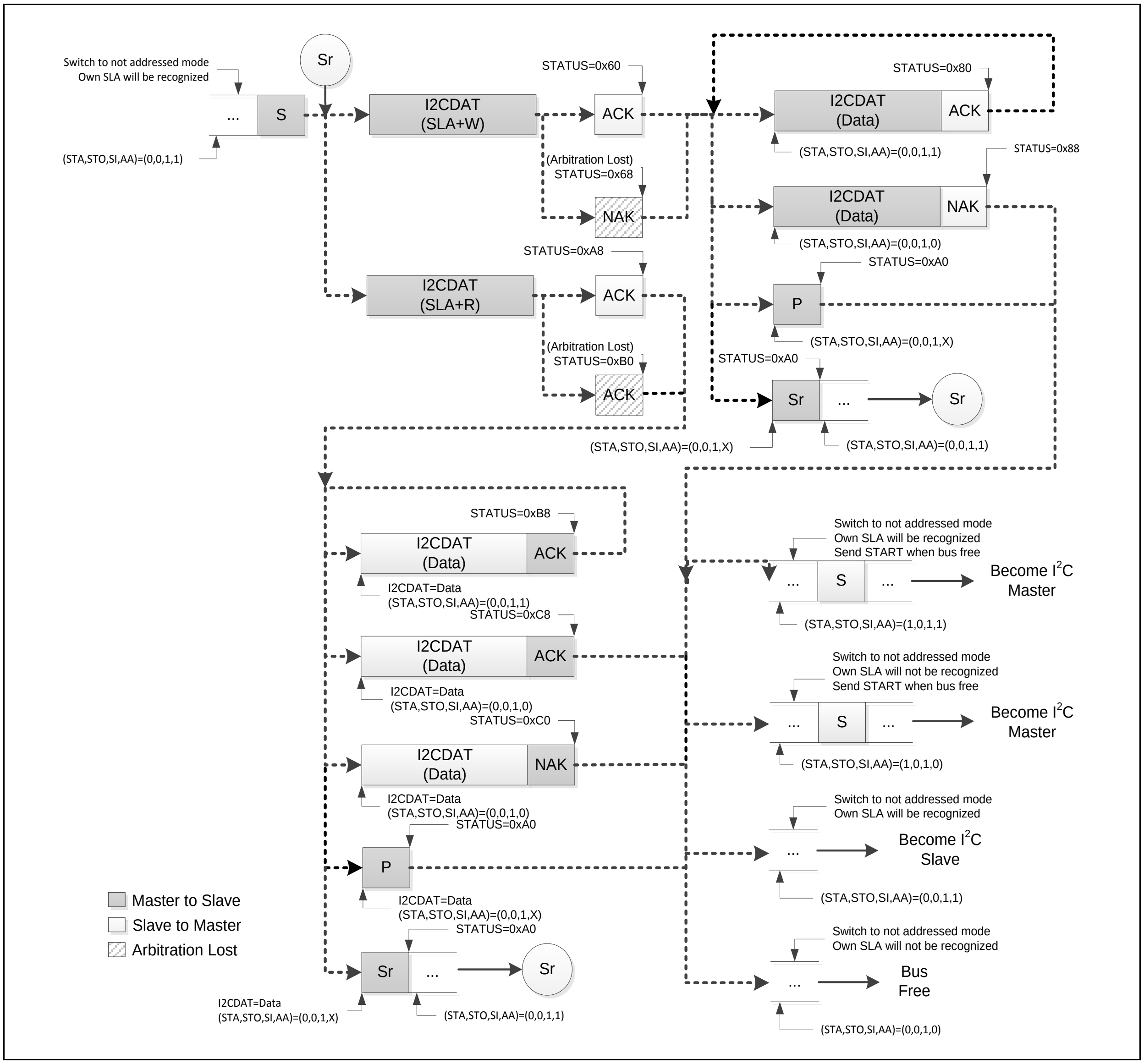

Switch to not addressed mode
Own SLA will be recognized
...
S
Sr
(STA,STO,SI,AA)=(0,0,1,1)
STATUS=0x60
I2CDAT
(SLA+W)
ACK
(Arbitration Lost)
STATUS=0x68
NAK
STATUS=0xA8
I2CDAT
(SLA+R)
ACK
(Arbitration Lost)
STATUS=0xB0
ACK
STATUS=0x80
I2CDAT
(Data)
ACK
(STA,STO,SI,AA)=(0,0,1,1)
STATUS=0x88
I2CDAT
(Data)
NAK
(STA,STO,SI,AA)=(0,0,1,0)
STATUS=0xA0
P
(STA,STO,SI,AA)=(0,0,1,X)
STATUS=0xA0
Sr
...
Sr
(STA,STO,SI,AA)=(0,0,1,X)
(STA,STO,SI,AA)=(0,0,1,1)
STATUS=0xB8
I2CDAT
(Data)
ACK
I2CDAT=Data
(STA,STO,SI,AA)=(0,0,1,1)
STATUS=0xC8
I2CDAT
(Data)
ACK
I2CDAT=Data
(STA,STO,SI,AA)=(0,0,1,0)
STATUS=0xC0
I2CDAT
(Data)
NAK
I2CDAT=Data
(STA,STO,SI,AA)=(0,0,1,0)
STATUS=0xA0
P
I2CDAT=Data
(STA,STO,SI,AA)=(0,0,1,X)
STATUS=0xA0
Sr
...
Sr
I2CDAT=Data
(STA,STO,SI,AA)=(0,0,1,X)
(STA,STO,SI,AA)=(0,0,1,1)
Switch to not addressed mode
Own SLA will be recognized
Send START when bus free
... S ...
Become I^{2}C Master
(STA,STO,SI,AA)=(1,0,1,1)
Switch to not addressed mode
Own SLA will not be recognized
Send START when bus free
... S ...
Become I^{2}C Master
(STA,STO,SI,AA)=(1,0,1,0)
Switch to not addressed mode
Own SLA will be recognized
...
Become I^{2}C Slave
(STA,STO,SI,AA)=(0,0,1,1)
Switch to not addressed mode
Own SLA will not be recognized
...
Bus Free
(STA,STO,SI,AA)=(0,0,1,0)
Master to Slave
Slave to Master
Arbitration Lost

Figure 6-84 Slave Mode Control Flow

If I^{2}C is still receiving data in addressed Slave mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0x88 as shown in the above figure when getting 0xA0 status.

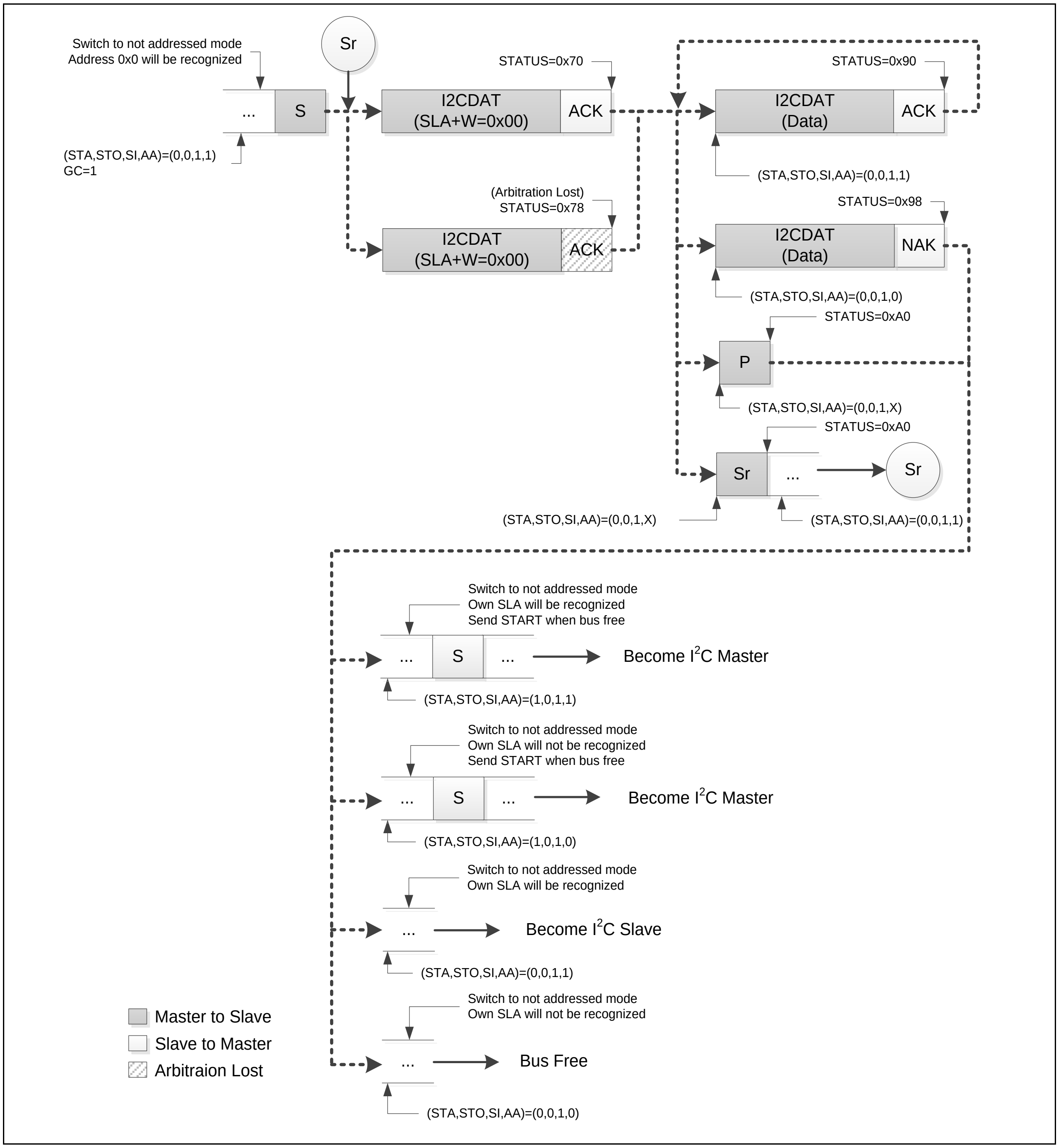
If I^{2}C is still transmitting data in addressed Slave mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0xC8 as shown in the above figure when getting 0xA0 status.

Note: After slave gets status of 0x88, 0xC8, 0xC0 and 0xA0, slave can switch to not address mode and own SLA will not be recognized. If entering this status, slave will not receive any I^{2}C signal or address from master. At this status, I^{2}C should be reset to leave this status.

6.16.4.20 General Call (GC) Mode

If the GCALL (I2CSADDRn [0]) bit is set to 1, the I^{2}C port hardware will respond to General Call address (00H). User can clear GCALL bit to disable general call function. When the GCALL bit is set and the I^{2}C is in Slave mode, it can receive the general call address by 0x00 after master send general call address to I^{2}C bus, then it will follow status of GCALL mode.

Note: (STA, STO, SI, AA) = (START, STOP, INTSTS, ACK)



Switch to not addressed mode
Address 0x0 will be recognized
Sr
...
S
(STA,STO,SI,AA)=(0,0,1,1)
GC=1
STATUS=0x70
I2CDAT
(SLA+W=0x00)
ACK
(Arbitration Lost)
STATUS=0x78
I2CDAT
(SLA+W=0x00)
ACK
STATUS=0x90
I2CDAT
(Data)
ACK
(STA,STO,SI,AA)=(0,0,1,1)
STATUS=0x98
I2CDAT
(Data)
NAK
(STA,STO,SI,AA)=(0,0,1,0)
STATUS=0xA0
P
(STA,STO,SI,AA)=(0,0,1,X)
STATUS=0xA0
Sr
...
Sr
(STA,STO,SI,AA)=(0,0,1,X)
(STA,STO,SI,AA)=(0,0,1,1)
Switch to not addressed mode
Own SLA will be recognized
Send START when bus free
...
S
...
Become I²C Master
(STA,STO,SI,AA)=(1,0,1,1)
Switch to not addressed mode
Own SLA will not be recognized
Send START when bus free
...
S
...
Become I²C Master
(STA,STO,SI,AA)=(1,0,1,0)
Switch to not addressed mode
Own SLA will be recognized
...
Become I²C Slave
(STA,STO,SI,AA)=(0,0,1,1)
Switch to not addressed mode
Own SLA will not be recognized
...
Bus Free
(STA,STO,SI,AA)=(0,0,1,0)
Master to Slave
Slave to Master
Arbitraion Lost

Figure 6-85 GC Mode

If I^2C is still receiving data in GCALL mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0x98 in the above figure when getting 0xA0 status.

Note: After slave gets status of 0x98 and 0xA0, slave can switch to not address mode and own SLA will not be recognized. If entering this status, slave will not receive any I^2C signal or address from master. At this time, I^2C controller should be reset to leave this status.

.

6.16.4.21 Multi-Master

In some applications, there are two or more masters on the same I^2C bus to access slaves, and the masters may transmit data simultaneously. The I^2C supports multi-master by including collision detection and arbitration to prevent data corruption.

- When I2CSTATUS = 0x38, an “Arbitration Lost” is received. Arbitration lost event maybe occur during the send START bit, data bits or STOP bit. User could set (START, STOP, INTSTS, ACK) = (1, 0, 1, X) to send START again when bus free, or set (START, STOP, INTSTS, ACK) = (0, 0, 1, X) to send STOP to back to not addressed Slave mode.
- When I2CSTATUS = 0x00, a “Bus Error” is received. To recover I2C bus from a bus error, STO should be set and SI should be cleared, and then STO is cleared to release bus.
  - Set (START, STOP, INTSTS, ACK) = (0, 1, 1, X) to stop current transfer
  - Set (START, STOP, INTSTS, ACK) = (0, 0, 1, X) to release bus

6.16.4.22 Example for Random Read on EEPROM

The following steps are used to configure the I^2C related registers when using I^2C to read data from EEPROM.

1. Set the multi-function pin in the “GPA_MFP” register as SCL and SDA pins.
2. Enable I^2C APB clock, I2C0ON (CLK_APBC[8]) = 1.
3. Set I2C0_RST (IPRSTC2[8]) =1 to reset I^2C controller then set I^2C controller to normal operation.
4. Set IPEN (I2CON[0]) = 1 to enable I^2C controller.
5. Give I^2C clock a divided register value for I^2C clock rate in the “I2CDIV”.
6. Set SETENA=0x00040000 in the “NVIC_ISER” register to set I^2C IRQ.
7. Set INTEN (I2CON[7]) = 1 to enable I^2C Interrupt.
8. Set I^2C slave address registers which are “I2CSADDR0~I2CADDR1”.

Random read operation is one of the methods of access EEPROM. The method allows the master to access any address of EEPROM space. The following figure shows the EEPROM random read operation.

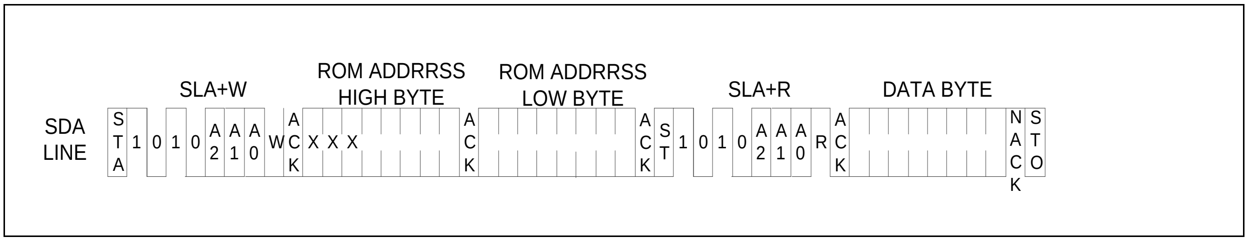


Figure 6-86 EEPROM Random Read

The following figure shows how to use I^{2}C controller to implement the protocol of EEPROM random read.

Note: (STA, STO, SI, AA) = (START, STOP, INTSTS, ACK)

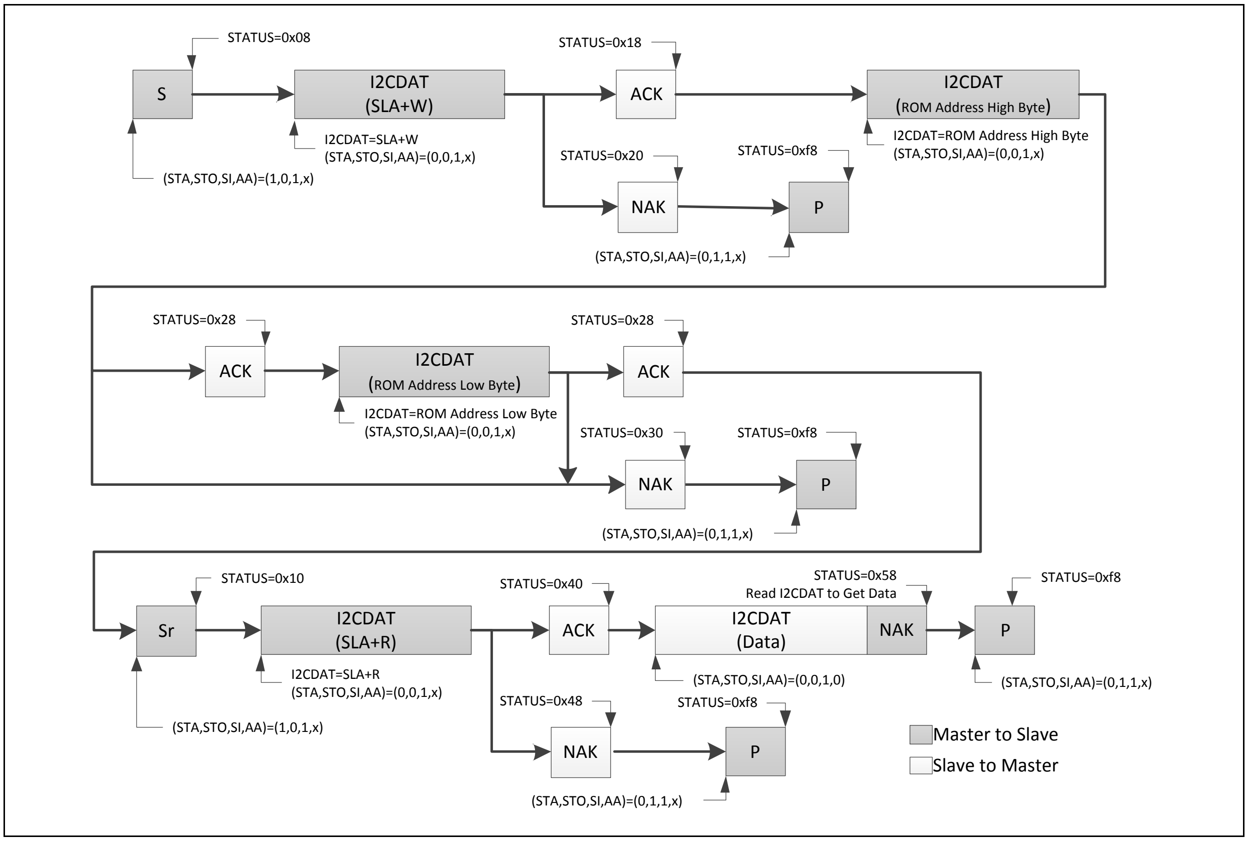

STATUS=0x08
S
I2CDAT (SLA+W)
I2CDAT=SLA+W
(STA,STO,SI,AA)=(0,0,1,x)
(STA,STO,SI,AA)=(1,0,1,x)
STATUS=0x18
ACK
I2CDAT (ROM Address High Byte)
I2CDAT=ROM Address High Byte
(STA,STO,SI,AA)=(0,0,1,x)
STATUS=0x20
NAK
STATUS=0xf8
P
(STA,STO,SI,AA)=(0,1,1,x)
STATUS=0x28
ACK
I2CDAT (ROM Address Low Byte)
I2CDAT=ROM Address Low Byte
(STA,STO,SI,AA)=(0,0,1,x)
STATUS=0x28
ACK
STATUS=0x30
NAK
STATUS=0xf8
P
(STA,STO,SI,AA)=(0,1,1,x)
STATUS=0x10
Sr
I2CDAT (SLA+R)
I2CDAT=SLA+R
(STA,STO,SI,AA)=(0,0,1,x)
(STA,STO,SI,AA)=(1,0,1,x)
STATUS=0x40
ACK
STATUS=0x58
Read I2CDAT to Get Data
I2CDAT (Data)
NAK
(STA,STO,SI,AA)=(0,0,1,0)
STATUS=0xf8
P
(STA,STO,SI,AA)=(0,1,1,x)
STATUS=0x48
NAK
STATUS=0xf8
P
(STA,STO,SI,AA)=(0,1,1,x)
▨ Master to Slave
□ Slave to Master

Figure 6-87 Protocol of EEPROM Random Read

The I^{2}C controller sends START to bus to be a master. Then it sends a SLA+W (Slave address + Write bit) to EERPOM followed by two bytes data address to set the EEPROM address to read. Finally, a Repeat START followed by SLA+R is sent to read the data from EEPROM.

6.16.5 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
I2C Base Address: **I2Cx_BA = 0x4002_0000+0x100000*x** **x=0,1**				
I2CON	I2Cx_BA+0x00	R/W	I^2C Control Register	0x0000_0000
I2CINTSTS	I2Cx_BA+0x04	R/W	I^2C Interrupt Status Register	0x0000_0000
I2CSTATUS	I2Cx_BA+0x08	R	I^2C Status Register	0x0000_00F8
I2CDIV	I2Cx_BA+0x0C	R/W	I^2C clock divided Register	0x0000_0000
I2CTOUT	I2Cx_BA+0x10	R/W	I^2C Time-out control Register	0x0000_0000
I2CDATA	I2Cx_BA+0x14	R/W	I^2C DATA Register	0x0000_0000
I2CSADDR0	I2Cx_BA+0x18	R/W	I^2C Slave address Register0	0x0000_0000
I2CSADDR1	I2Cx_BA+0x1C	R/W	I^2C Slave address Register1	0x0000_0000
I2CSAMASK0	I2Cx_BA+0x28	R/W	I^2C Slave address Mask Register0	0x0000_0000
I2CSAMASK1	I2Cx_BA+0x2C	R/W	I^2C Slave address Mask Register1	0x0000_0000
I2CON2	I2Cx_BA+0x3C	R/W	I^2C Control Register 2	0x0000_0000
I2CSTATUS2	I2Cx_BA+0x40	R	I^2C Status Register 2	0x0000_0000

6.16.6 Register Description

I^{2}C Control Register (I2CON)

Register	Offset	R/W	Description	Reset Value
I2CON	I2Cx_BA+0x00	R/W	I^{2}C Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTEN	**Reserved**		**I2C_STS**	**START**	**STOP**	**ACK**	**IPEN**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**INTEN**	**Interrupt Enable Control** 0 = I^{2}C interrupt Disabled. 1 = I^{2}C interrupt Enabled.
[6:5]	**Reserved**	Reserved.
[4]	**I2C_STS**	**I^{2}C Status** When a new state is present in the I2CSTATUS register, if the INTEN bit is set, the I2C interrupt is requested. It must write one by software to this bit after the I2CINTSTS[0] is set to 1 and the I2C protocol function will go ahead until the STOP is active or the IPEN is disabled. 0 = I^{2}C's Status disabled and the I^{2}C protocol function will go ahead. 1 = I^{2}C's Status active.
[3]	**START**	**I^{2}C START Command** Setting this bit to 1 to enter Master mode, the device sends a START or repeat START condition to bus when the bus is free and it will be cleared to 0 after the START command is active and the STATUS has been updated. 0 = After START or repeat START is active. 1 = Sends a START or repeat START condition to bus.
[2]	**STOP**	**I^{2}C STOP Control Bit** In Master mode, set this bit to 1 to transmit a STOP condition to bus then the controller will check the bus condition if a STOP condition is detected and this bit will be cleared by hardware automatically. In Slave mode, set this bit to 1 to reset the controller to the defined "not addressed" Slave mode. This means it is NO LONGER in the slave receiver mode to receive data from the master transmit device. 0 = Will be cleared by hardware automatically if a STOP condition is detected. 1 = Sends a STOP condition to bus in Master mode or reset the controller to "not addressed" in Slave mode.

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

Bits	Description	
[1]	**ACK**	**Assert Acknowledge Control Bit** 0 = When this bit is set to 0 prior to address or data received, a Not acknowledged (high level to SDA) will be returned during the acknowledge clock pulse. 1 = When this bit is set to 1 prior to address or data received, an acknowledged will be returned during the acknowledge clock pulse on the SCL line when (a). A slave is acknowledging the address sent from master. (b). The receiver devices are acknowledging the data sent by transmitter.
[0]	**IPEN**	**I^{2}C Function Enable Control** 0 = I^{2}C function Disabled. 1 = I^{2}C function Enabled.

I^{2}C Interrupt Status Register (I2CINTSTS)

Register	Offset	R/W	Description	Reset Value
I2CINTSTS	I2Cx_BA+0x04	R/W	I^{2}C Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WAKEUP_ACK_DONE	**Reserved**					**TIF**	**INTSTS**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**WAKEUP_ACK_DONE**	**Wake-up Address Frame Acknowledge Bit Done** 0 = The ACK bit cycle of address match frame isn't done. 1 = The ACK bit cycle of address match frame is done in power-down.
[6:2]	**Reserved**	Reserved.
[1]	**TIF**	**Time-out Status** 0 = No Time-out flag. Software can cleat this flag. 1 = Time-Out flag active and it is set by hardware. It can interrupt CPU when INTEN bit is set.
[0]	**INTSTS**	**I^{2}C STATUS's Interrupt Status** 0 = No bus event occurred. 1 = New state is presented in the I2CSTATUS. Software can write 1 to cleat this bit.

I^{2}C Status Register (I2CSTATUS)

Register	Offset	R/W	Description	Reset Value
I2CSTATUS	I2Cx_BA+0x08	R	I^{2}C Status Register	0x0000_00F8

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STATUS							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**STATUS**	**I^{2}C Status Bits (Read Only)** Indicates the current status code of the bus information. The detail information about the status is described in the sections of I^{2}C protocol register and operation mode**.**

I^{2}C Baud Rate Control Register (I2CDIV)

Register	Offset	R/W	Description	Reset Value
I2CDIV	I2Cx_BA+0x0C	R/W	I^{2}C clock divided Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CLK_DIV							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**CLK_DIV**	**I2C Clock Divided Bits** The I^{2}C clock rate bits: Data Baud Rate of I^{2}C = PCLK /(4 x (CLK_DIV + 1)). **Note:** the minimum value of CLK_DIV is 4.

I²C Time-out Counter Register (I2CTOUT)

Register	Offset	R/W	Description	Reset Value
I2CTOUT	I2Cx_BA+0x10	R/W	I²C Time-out control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**DIV4**	**TOUTEN**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**DIV4**	**Time-out Counter Input Clock Divider by 4** 0 = Disabled. 1 = Enabled. When Enabled, the time-out period is extended 4 times.
[0]	**TOUTEN**	**Time-out Counter Enable/Disable Control** 0 = Disabled. 1 = Enabled. When set this bit to enable, the 14 bits time-out counter will start counting when INTSTS (I2CINTSTS[0]) is cleared. Setting flag STAINTSTS to high or the falling edge of I²C clock or stop signal will reset counter and re-start up counting after INTSTS is cleared.

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

I^{2}C Data Register (I2CDATA)

Register	Offset	R/W	Description	Reset Value
I2CDATA	I2Cx_BA+0x14	R/W	I^{2}C DATA Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**DATA**	**I^{2}C Data Bits** The DATA contains a byte of serial data to be transmitted or a byte which has just been received. **Note:** Refer to Data register section for more detail information.

I^{2}C Slave Address Register (I2CSADDRx)

Register	Offset	R/W	Description	Reset Value
I2CSADDR0	I2Cx_BA+0x18	R/W	I^{2}C Slave address Register0	0x0000_0000
I2CSADDR1	I2Cx_BA+0x1C	R/W	I^{2}C Slave address Register1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SADDR							GCALL

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:1]	**SADDR**	**I^{2}C Salve Address Bits** The content of this register is irrelevant when the device is in Master mode. In the Slave mode, the seven most significant bits must be loaded with the device's own address. The device will react if either of the address is matched.
[0]	**GCALL**	**General Call Function** 0 = General Call Function Disabled. 1 = General Call Function Enabled. **Note:** Refer to Address Register section for more detail information..

I2C SLAVE ADDRESS MASK REGISTER (I2CSAMASKx)

Register	Offset	R/W	Description	Reset Value
I2CSAMASK0	I2Cx_BA+0x28	R/W	I^{2}C Slave address Mask Register0	0x0000_0000
I2CSAMASK1	I2Cx_BA+0x2C	R/W	I^{2}C Slave address Mask Register1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SAMASK							**Reserved**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:1]	**SAMASK**	**I^{2}C Slave Address Mask Bits** 0 = Mask disable (the received corresponding register bit should be exact the same as address register). 1 = Mask enable (the received corresponding address bit is don't care)..
[0]	**Reserved**	Reserved.

I^{2}C Control Register 2 (I2CON2)

Register	Offset	R/W	Description	Reset Value
I2CON2	I2Cx_BA+0x3C	R/W	I^{2}C Control Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**NOSTRETCH**	**TWOFF_EN**	**Reserved**	**UNDER_INTEN**	**OVER_INTEN**	**WKUPEN**

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**NOSTRETCH**	**NO STRETCH the I^{2}C BUS** 0 = The I^{2}C SCL bus is stretched by hardware if the INTSTS (I2CINTSTS[0]) is not cleared in master mode. 1 = The I^{2}C SCL bus is not stretched by hardware if the INTSTS is not cleared in master mode.
[4]	**TWOFF_EN**	**TWO LEVEL FIFO Enable Control** 0 = Disabled. 1 = Enabled.
[3]	**Reserved**	Reserved.
[2]	**UNDER_INTEN**	**I^{2}C UNDER RUN Interrupt Control Bit** 0 = Under run event interrupt Disabled. 1 = Send a interrupt to system when the TWOFF bit is enabled and there is under run event happened in transmitted fifo.
[1]	**OVER_INTEN**	**I^{2}C OVER RUN Interrupt Control Bit** 0 = Overrun event interrupt Disabled. 1 = Send a interrupt to system when the TWOFF bit is enabled and there is over run event in received fifo.
[0]	**WKUPEN**	**I^{2}C Wake-up Function Enable Control** 0 = I^{2}C wake-up function Disabled. 1 = I^{2}C wake-up function Enabled.

I^{2}C Status Register 2 (I2CSTATUS2)

Register	Offset	R/W	Description	Reset Value
I2CSTATUS2	I2Cx_BA+0x40	R	I^{2}C Status Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	BUS_FREE	EMPTY	FULL	WR_STATUS	UNDERUN	OVERUN	WKUPIF

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**BUS_FREE**	**Bus Free Status** The bus status in the controller. 0 = I^{2}C's "Start" condition is detected on the bus. 1 = Bus free and it is released by "STOP" condition or the controller is disabled.
[5]	**EMPTY**	**I^{2}C TWO LEVEL FIFO EMPTY** 0 = RX FIFO no empty when the TWOFF_EN = 1. 1 = RX FIFO empty when the TWOFF_EN = 1.
[4]	**FULL**	**I^{2}C TWO LEVEL FIFO FULL** 0 = TX FIFO no full when the TWOFF_EN = 1. 1 = TX FIFO full when the TWOFF_EN = 1.
[3]	**WR_STATUS**	**I^{2}C Read/Write Status Bit in Address Wake-up Frame** 0 = Write command be record on the address match wake-up frame. 1 = Read command be record on the address match wake-up frame.
[2]	**UNDERUN**	**I^{2}C UNDER RUN Status Bit** 0 = The transmitted FIFO is not under run when the TWOFF_EN = 1. 1 = The transmitted FIFO is under run when the TWOFF_EN = 1.
[1]	**OVERUN**	**I^{2}C OVER RUN Status Bit** 0 = The received FIFO is not over run when the TWOFF_EN = 1. 1 = The received FIFO is over run when the TWOFF_EN = 1.
[0]	**WKUPIF**	**Wake-up Interrupt Flag** 0 = Wake-up flag inactive. 1 = Wake-up flag active. Software can write 1 to clear this flag

6.17 SPI

6.17.1 Overview

The Serial Peripheral Interface (SPI) is a synchronous serial data communication protocol. Devices communicate in Master/Slave mode with 4-wire bi-direction interface. It is used to perform a serial-to-parallel conversion on data received from a peripheral device, and a parallel-to-serial conversion on data transmitted to a peripheral device. The SPI controller can be configured as a master or a slave device.

The SPI controller supports wake-up function. When this chip stays in Power-down mode, it can be waked up by off-chip device.

This controller supports variable serial clock function for special application and 2-bit transfer mode to connect 2 off-chip slave devices. The SPI controller also supports PDMA function to access the data buffer.

6.17.2 Features

- Up to two sets of SPI controllers
- Supports Master (max. 32 MHz) or Slave (max. 16 MHz) mode operation
- Supports 1 bit and 2 bit transfer mode
- Support Dual IO transfer mode
- Configurable bit length of a transaction from 8 to 32-bit
- Supports MSB first or LSB first transfer sequence
- Two slave select lines supported in Master mode
- Configurable byte or word suspend mode
- Supports byte re-ordering function
- Supports variable serial clock in Master mode
- Provide separate 8-level depth transmit and receive FIFO buffer
- Supports wake-up function
- Supports PDMA transfer
- Supports 3-wires, no slave select signal, bi-direction interface

6.17.3 Block Diagram

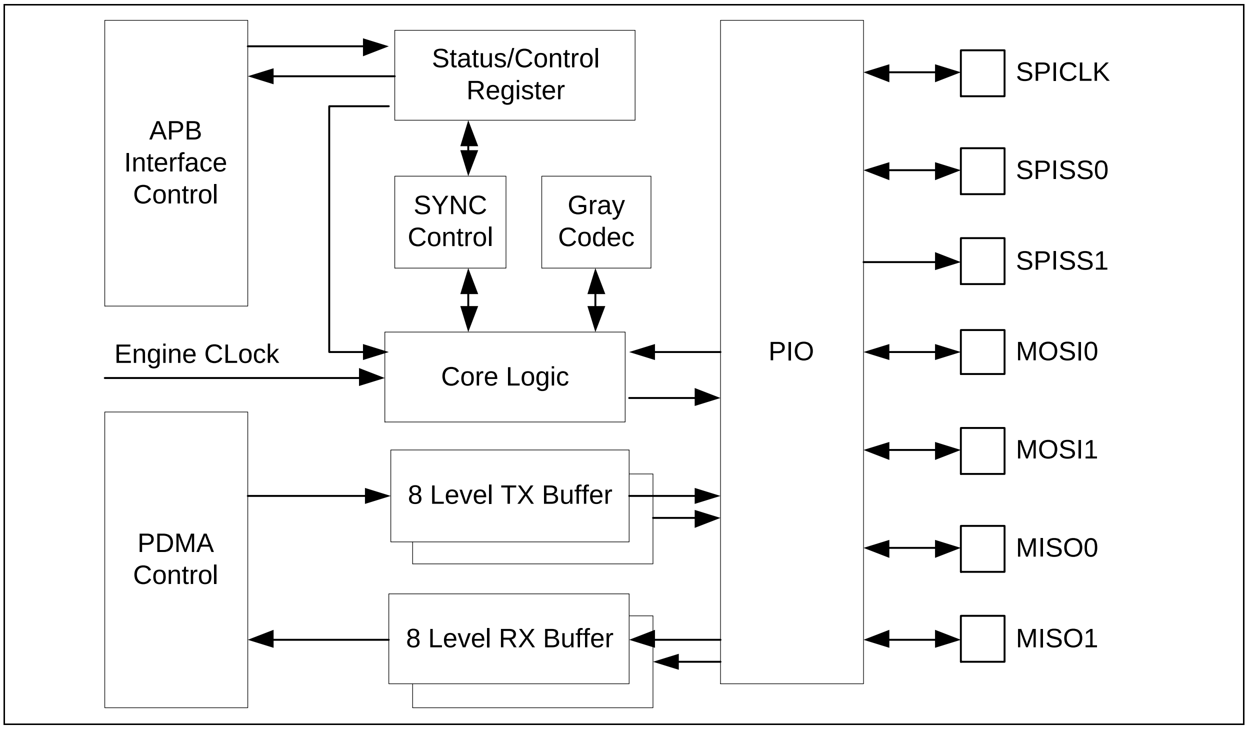


Figure 6-88 SPI Block Diagram

6.17.4 Basic Configuration

- The SPI pin functions are configured in GPA_MFP/ GPB_MFP and GPE_MFP register.
- The SPI peripheral clock can be enabled in SPI_EN(APBCLK[13:12]) bit and its source can be selected in SPI_S(CLKSEL2[21:20]) bit.

6.17.5 Functional Description

6.17.5.1 SPI Peripheral Clock and Serial Clock

SPI controller needs the SPI peripheral clock to drive the SPI logic unit to perform the data transfer. The SPI peripheral clock rate is determined by the settings of clock source and clock divisor. The SPIx_S bit of CLKSEL2 register determines the clock source of the SPI peripheral clock. The clock source can be HCLK or PLL output clock. The DIVIDER setting of SPI_DIVIDER register determines the divisor of the clock rate calculation.

In master mode, if the variable clock function is disabled, VARCLK_EN (SPI_CTL[23]) = 0, the output frequency of the serial clock output pin is equal to the SPI peripheral clock rate. In slave mode, the SPI serial clock is provided by an off-chip master device. The frequency of SPI peripheral clock in slave device must be faster than the serial clock rate of the master device connected together. The frequency of SPI peripheral clock cannot be faster than the APB clock rate regardless of master mode or slave mode.

6.17.5.2 Master/Slave Mode

This SPI controller can be set as Master or Slave mode by setting the SLAVE bit (SPI_CTL[18]) to communicate with the off-chip SPI slave or master device. The application block diagrams in Master and Slave mode are shown below.

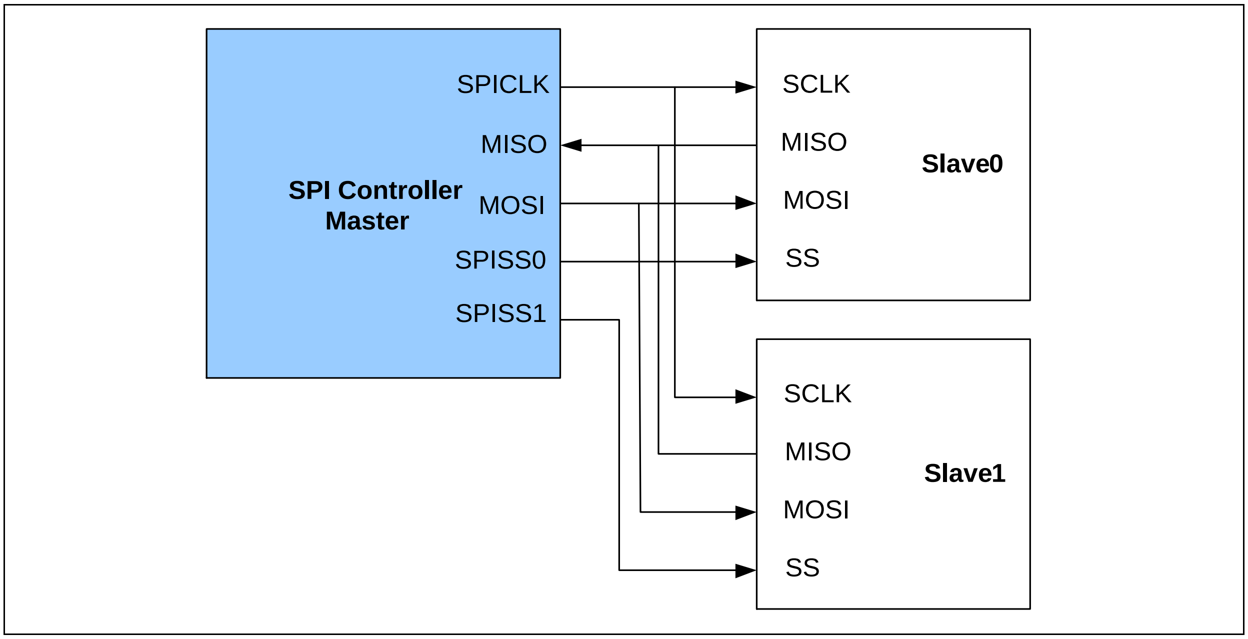


Figure 6-89 SPI Master Mode Application Block Diagram

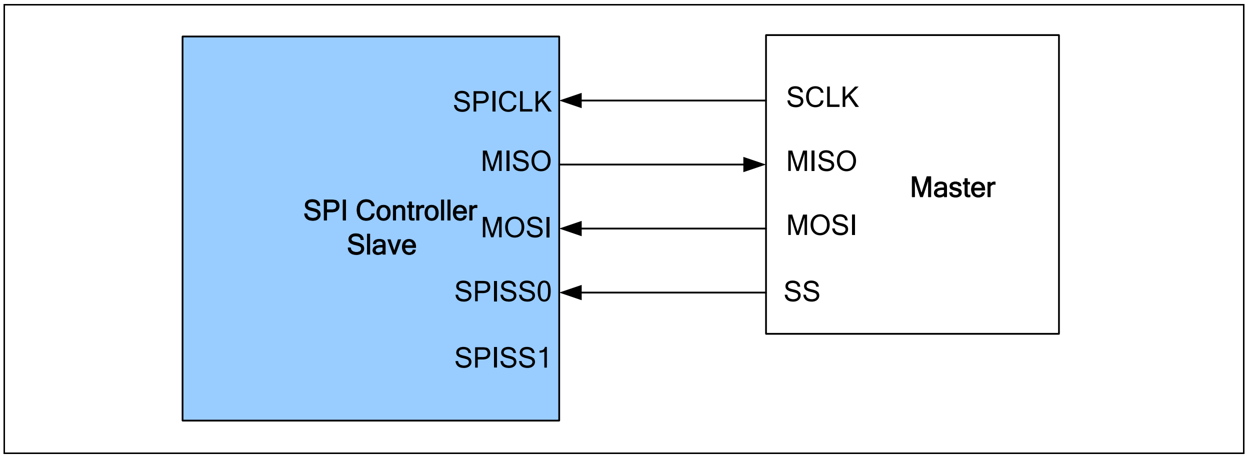


Figure 6-90 SPI Slave Mode Application Block Diagram

6.17.5.3 Slave Selection

In Master mode, this SPI controller can drive up to two off-chip slave devices through the slave select output signals SPISS0 and SPISS1, but it is a time-sharing operation and it can not operate with two slave devices simultaneously.

In Slave mode, the off-chip master device drives the slave select signal from the SPISS0 port to this SPI controller. In Master/Slave mode, the active level of slave select signal can be programmed to low active or high active in SS_LVL bit (SPI_SSR[2]), and the SS_LTRIG bit (SPI_SSR[4]) define the slave select signal SPIxSS0/1 is level trigger or edge trigger. The selection of trigger condition depends on what type of peripheral slave/master device is connected.

In Slave mode, if the SS_LTRIG bit is configured as level trigger, the LTRIG_FLAG bit (SPI_SSR[4]) is used to indicate if the count of received bits among one transaction meets the requirement which define in TX_BIT_LEN.

6.17.5.4 Level-trigger / Edge-trigger

In Slave mode, the slave select signal can be configured as level-trigger, SS_LTRIG (SPI_SSR[4]) =1 , or edge-trigger, SS_LTRIG = 0. In edge-trigger, the data transfer starts from an active edge and ends on an inactive edge. If master does not send an inactive edge to slave, the transfer procedure will not be completed and the unit transfer interrupt flag of slave will not be set. In level-trigger, the following two conditions will terminate the transfer procedure and the unit transfer interrupt flag of slave will be set. The first condition is that if the number of transferred bits matches the settings of TX_BIT_LEN (SPI_CTL[7:3]), the unit transfer interrupt flag of slave will be set. The second condition, if master set the slave select pin to inactive level during the transfer is in progress, it will force slave device to terminate the current transfer no matter how many bits have been transferred and the unit transfer interrupt flag will be set. User can read the status of LTRIG_FLAG (SPI_STATUS[4]) bit to check if the data has been completely transferred.

6.17.5.5 Automatic Slave Select

In Master mode, if the AUTOSS bit (SPI_SSR[3]) is set as 1, the slave select signals will be generated automatically and output to SPIxSS0 and SPIxSS1 ports according to SSR[0] (SPI_SSR[0]) and SSR[1] (SPI_SSR[1]) whether it is enabled or not. It means that the slave select signals, which is enabled in SPI_SSR[1:0] register is asserted by the SPI controller when the SPI data transfer is started by setting the GO_BUSY bit (SPI_CTL[0]) and is de-asserted after the data transfer is finished. If the AUTOSS bit is cleared to 0, the slave select output signals are asserted and de-asserted by manual setting and clearing the related bits in SPI_SSR[1:0] register. The active level of the slave select output signals is specified in SS_LVL bit (SPI_SSR[2]).

In master mode, if the value of SP_CYCLE (SPI_CTL[15:12]) is less than 3 and the AUTOSS (SPI_SSR[3]) is set as 1, the slave select signal will keep at active state between two successive transactions.

In slave mode, to recognize the inactive state of the slave select signal, the inactive period of the slave select signal must be larger than or equal to 6 peripheral clock periods between two successive transactions.

6.17.5.6 No Slave Select Mode (3-WIRE mode)

When the software sets the NOSLVSEL (SPI_SSR[5]) bit to enable the 3-wire mode, the SPI controller can work with no slave select signal in slave mode. The NOSLVSEL bit only takes effect in slave mode. It only needs three pins, SPICLK, SPI_MISO, and SPI_MOSI, to communicate with a SPI master. The SPISS pin can be configured as a GPIO. When the NOSLVSEL bit is set to 1, the SPI slave will be ready to transmit/receive data after the GO_BUSY (SPI_CTL[0]) bit is set to 1. In 3-wire mode, the SS_LTRIG, SPI_SSR[4], shall be set as 1.

In normal operation, the interrupt flag in SLV_START_INTSTS (SPI_STATUS[6]) will be set when the transfer has start if the SSTA_INTEN (SPI_SSR[9]) is set 1. The serial clock toggle input is the transfer start condition. And there is also interrupt event when the received data meet the required bits which define in TX_BIT_LEN (SPI_CTL[7:3]). If the received bits are less than the requirement and there is no more serial clock input over the time period which is defined by the user in Slave mode with no slave select, the user can set the SLV_ABORT (SPI_SSR[8]) bit to force the current transfer done and then the user can get a transfer done interrupt event.

6.17.5.7 Variable Clock Function

In master mode, if the VARCLK_EN bit (SPI_CTL[23]) is set to 1, the output of serial clock can be programmed as variable frequency pattern. The serial clock period of each cycle depends on the setting of the SPI_VARCLK register. When the variable clock function is enabled, the TX_BIT_LEN setting must be set as 0x10 to configure the data transfer as 16-bit transfer mode. The VARCLK[31] determines the clock period of the first clock cycle. If VARCLK[31] is 0, the first clock cycle depends on the DIVIDER1 setting; if it is 1, the first clock cycle depends on the DIVIDER2 setting. Two successive bits in VARCLK[30:1] defines one clock cycle. The bit field VARCLK[30:29] defines the second clock cycle of SPI serial clock of a transaction, and the bit field VARCLK[28:27] defines the third clock cycle and so on. The VARCLK[0] is unmeaning. The following figure shows the timing relationship among the serial clock (SPICLK), the VARCLK, the DIVIDER1 and the DIVIDER2 registers.

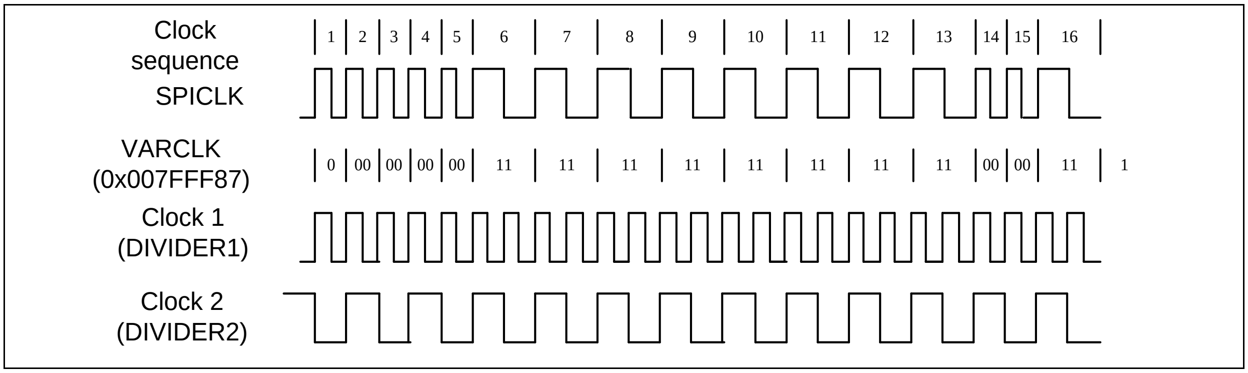


Figure 6-91 SPI Variable Clock Frequency

6.17.5.8 Clock Polarity

The CLKP bit (SPI_CTL[11]) defines the serial clock idle state in Master mode. If CLKP = 1, the output SPICLK is idle at high state. If CLKP = 0, it is idle at low state. For variable serial clock, it works in CLKP = 0 only.

6.17.5.9 Transmitting/Receiving Bit Length

The bit length of a transaction word is defined in TX_BIT_LEN bit (SPI_CTL[7:3]). It can be configured up to 32 bits in a transaction word for transmitting and receiving.

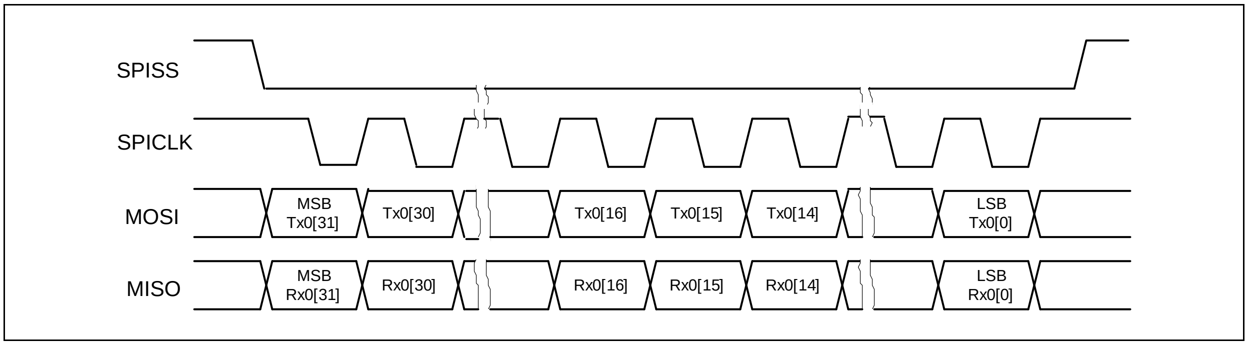


Figure 6-92 SPI 32-bit in One Transaction

6.17.5.10 LSB First

The LSB bit (SPI_CTL[10]) defines the bit transfer sequence in a transaction. If set the LSB bit to 1, the transfer sequence is LSB first. The bit 0 will be transferred firstly. If clear the LSB bit to 0, the transfer sequence is MSB first (refer to the figure above).

6.17.5.11 Edge Condition

The TX_NEG bit (SPI_CTL[2]) defines the data transmitted out either at negative edge or at positive edge of serial clock SPICLK.

The RX_NEG bit (SPI_CTL[1]) defines the data received in either at negative edge or at positive edge of serial clock SPICLK. Note that TX_NEG and RX_NEG must be exclusive.

6.17.5.12 Word Suspend

These four bits field of SP_CYCLE (SPI_CTL[15:12]) provide a configurable clock suspend interval 0.5 ~ 15.5 serial clock periods between two successive transaction words in Master mode. The definition of the suspend interval is the interval between the last clock edge of the preceding transaction word and the first clock edge of the following transaction word. The default value of SP_CYCLE is 0x3 (3.5 serial clock cycles). This SP_CYCLE setting will not take effect to the word suspend interval if the software disables the FIFO mode.

If both the VARCLK_EN(SPI_CTL[23]), and the FIFOM(SPI_CTL[21]), are set as 1, the minimum word suspend period is (6.5 + SP_CYCLE)*SPI serial clock period.

6.17.5.13 Byte Reorder

When the transfer is set as MSB first (LSB (SPI_CTL[10]) = 0), the REORDER (SPI_CTL[19]) is enabled, the data stored in the SPI_TX FIFO and SPI_RX FIFO will be rearranged in the order as [BYTE0, BYTE1, BYTE2, BYTE3] in 32 bit transfer mode (TX_BIT_LEN (SPI_CTL[7:3])= 0). The

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

sequence of transmitted/received data will be BYTE0, BYTE1, BYTE2, and then BYTE3. If the TX_BIT_LEN is set as 24-bits transfer mode, the data in TX buffer and RX buffer will be rearranged as [unknown byte, BYTE0, BYTE1, BYTE2]. The SPI controller will transmit/receive data with the sequence of BYTE0, BYTE1 and then BYTE2. Each byte will be transmitted/received with MSB first. The rule of 16-bits mode is the same as above. Byte reorder function is only available when TX_BIT_LEN is configured as 16, 24, and 32 bits.

Note: The byte reorder function is not supported when the variable serial clock function is enabled.

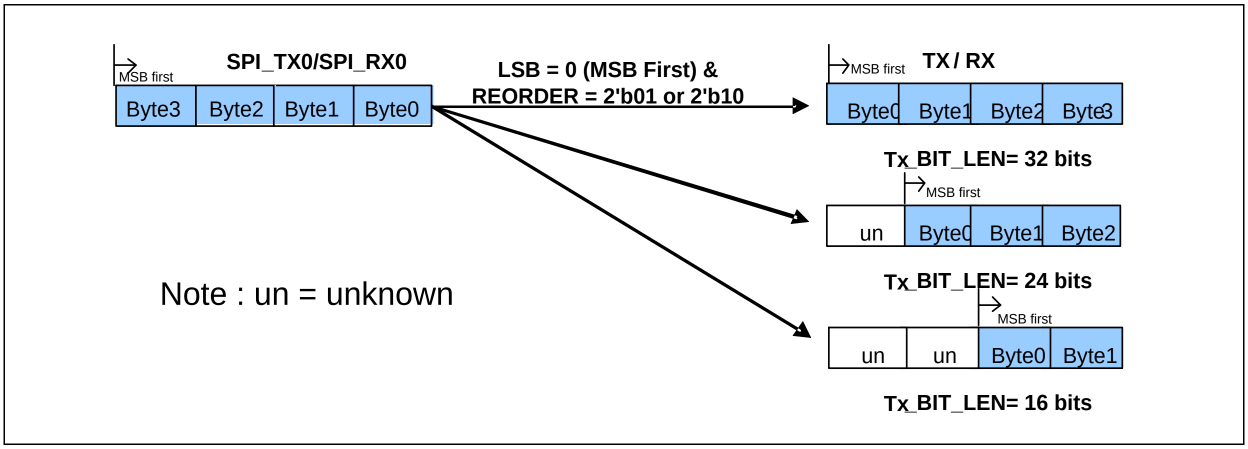


Figure 6-93 SPI Byte Reorder

6.17.5.14 Byte Suspend

In Master mode, if REORDER (SPI_CTL[19]) is set to 1, the hardware will insert a suspend interval of 0.5~15.5 serial clock periods between two successive bytes in a transfer word. Both settings of byte suspend interval and word suspend interval are configured in SP_CYCLE (SPI_CTL[15:12]). Byte Suspend is only used in SPI Byte Reorder mode.

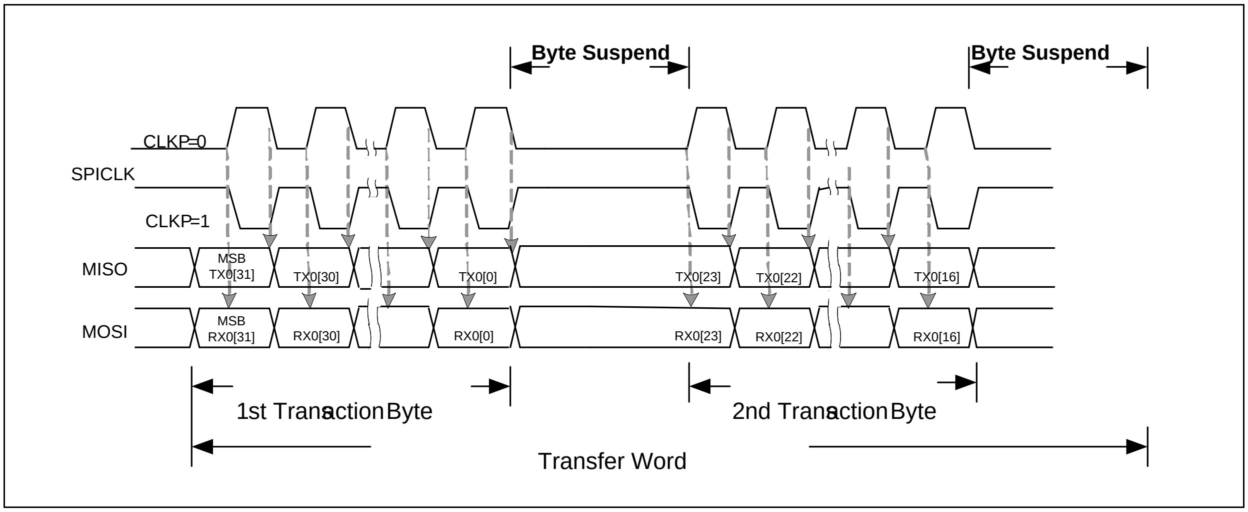


Figure 6-94 SPI Byte Suspend Mode

6.17.5.15 Interrupt

Each SPI controller can generates an individual interrupt when data transfer is finished or the FIFO reach the threshold level and the respective interrupt event flag INTSTS (SPI_STATUS[7], TX_INTSTS (SPI_STATUS[10]), RX_INTSTS (SPI_STATUS[8]) will be set. The interrupt event flag will generates an interrupt to CPU if the interrupt enable bit INTEN (SPI_CTL[17]) is set. The interrupt event flag INTSTS, TXINT_STS (SPI_STATUS[10]), and RXINT_STS (SPI_STATUS[8]) can be cleared only by writing 1 to it.

As the SPI controller finishes a unit transfer, the unit transfer interrupt flag INTSTS (SPI_STATUS[7]) will be set to 1. The unit transfer interrupt event will generate an interrupt to CPU if the unit transfer interrupt enable bit IE (SPI_CTL[17]) is set. The unit transfer interrupt flag can be cleared only by writing 1 to it.

In 3-wire mode, the slave 3-wire mode start interrupt flag, SLV_START_INTSTS (SPI_STATUS[6]), will be set to 1 when the slave senses the SPI clock signal. The SPI controller will issue an interrupt if the SSTA_INTEN (SPI_SSR[9]) is set to 1. If the count of the received bits is less than the setting of TX_BIT_LEN (SPI_CTL[7:3]) and there is no more serial clock input over the expected time period which is defined by the user, the user can set the SLV_ABORT bit to abort the current transfer. The unit transfer interrupt flag, INTSTS (SPI_STATUS[7]), will be set to 1 if the software set the SLV_ABORT (SPI_SSR[8]) bit.

In FIFO mode, there is time-out function to inform user. If there is a received data in the FIFO and it does not be read by software over 64 SPI peripheral clock periods in master mode or over 576 SPI peripheral clock periods in slave mode, it will send a time-out interrupt to the system if the time-out interrupt enable bit TIMEOUT_EN, FIFO_CTL[7], is set to 1.

6.17.5.16 Two Bit Transfer Mode

This SPI controller also supports 2-bit transfer mode when enabling the TWOB bit, SPI_CTL[22]. When the TWOB bit is enabled, it can transmit and receive 2-bit serial transfer data simultaneousl$^{y.}$

The 1st bit is through the MOSI0 and MISO0 port to transmit the data from SPI_TX0 register and receive the data into SPI_RX0 regist^{er}. The 2nd bit is through the MOSI1and MISO1 port to transmit the data from SPI_TX1 register and receive the data into SPI_RX1 register. The system block and timing of 2-bit transfer mode are shown below.

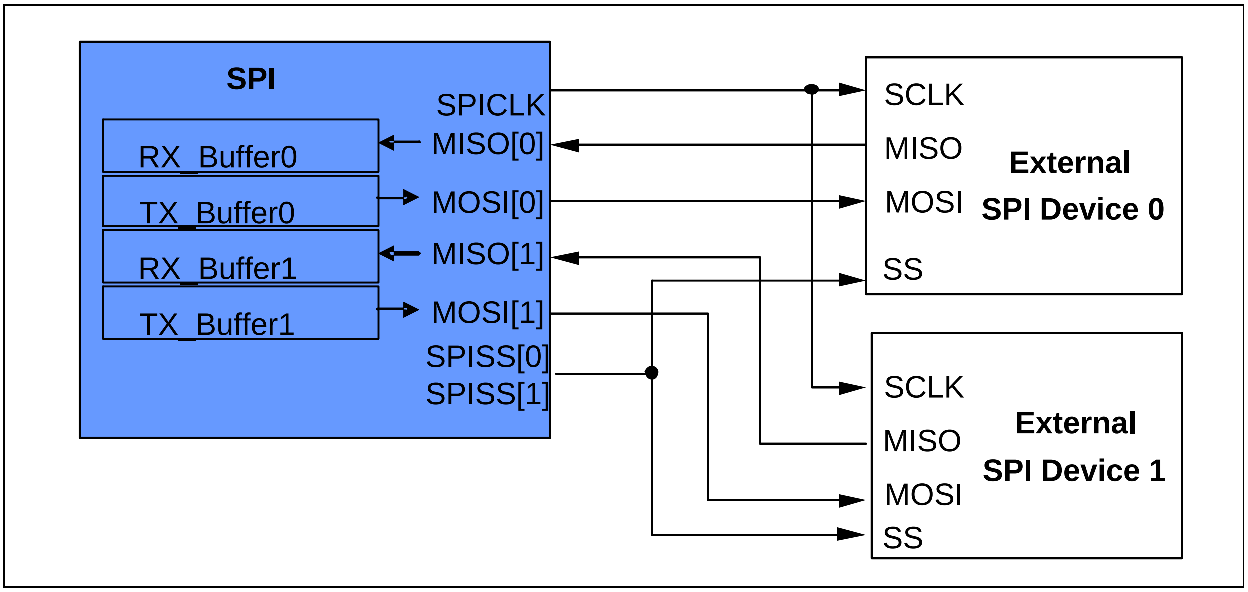


Figure 6-95 SPI 2-bit Transfer Mode

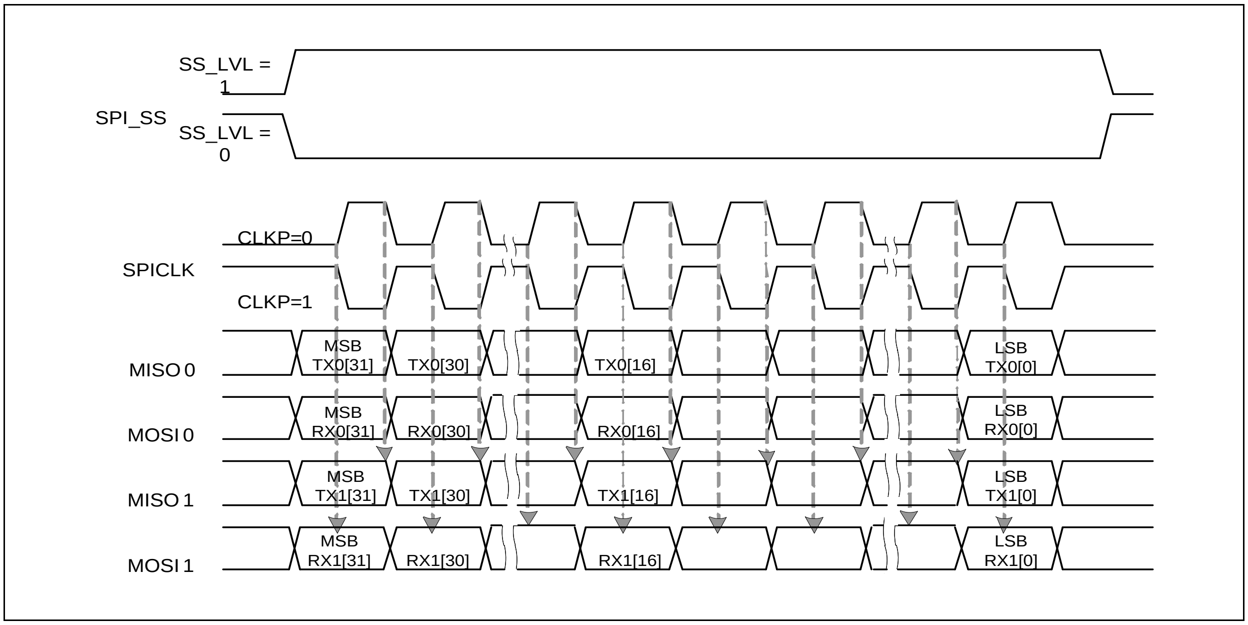


Figure 6-96 SPI 2-bit Transfer Mode Timing Diagram

6.17.5.17 Dual IO Mode

This SPI controller also supports dual IO transfer when set the DUAL_IO_EN bit (SPI_CTL[29]) to 1. The DUAL_IO_DIR bit (SPI_CTL[28] is used to define the direction of the transfer data. When set the DUAL_IO_DIR bit to 1, the controller will send the data to external device. When the DUAL_IO_DIR bit set 0, the controller will read the data from the external device. This function supports 8, 16, 24, and 32-bits of bit length.

The dual IO mode is not supported when the 3-wire mode or the byte reorder function is enabled.

If both the DUAL_IO_EN and DUAL_IO_DIR bits are set as 1, the MOSI0 is the even bit data output and the MISO0 will be set as the odd bit data output. If the DUAL_IO_EN is set as 1 and DUAL_IO_DIR is set as 0, both the MISO0 and MOSI0 will be set as data input ports.

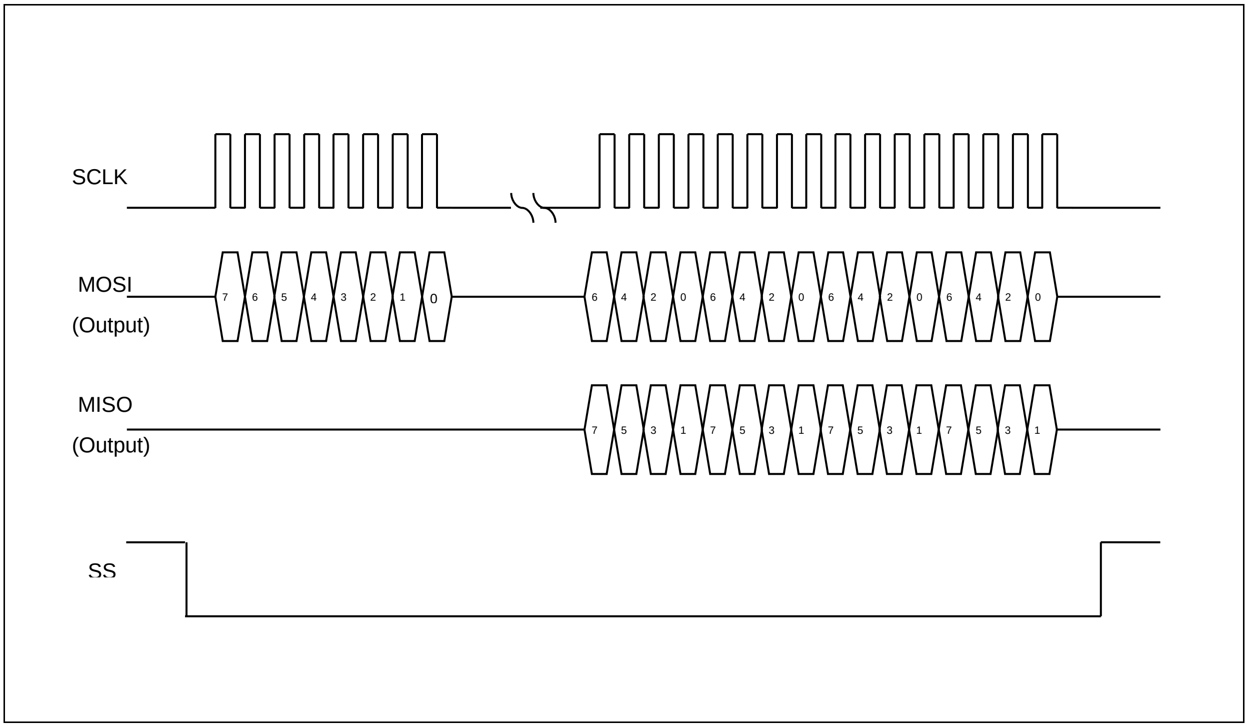


Figure 6-97 SPI DUAL-I/O output Sequence

Note: The byte reorder and no slave select functions are not support in DUAL I/O mode.

6.17.5.18 Time Out

In FIFO mode, there is time-out function to inform user. If there is a received data in the FIFO and it isn't read by user over 64 SPI peripheral clock in master mode or over 576 SPI peripheral clock in slave mode, it will send a time-out interrupt to the system if the time-out enable bit TIMEOUT_EN(SPI_FFCTL[7]) is set to 1

6.17.5.19 FIFO Mode

The SPI controller supports FIFO mode when the FIFOM bit(SPI_CTL[21]) is set as 1. The SPI controllers equip with 8 32-bit wide transmit and receive FIFO buffers.

The transmit FIFO buffer is an 8-level depth, 32-bit wide, first-in, first-out register buffer. The software can write data to the transmit FIFO buffer by writing the SPI_TX0 register. The data stored in the transmit FIFO buffer will be read and sent out by the transmission control logic. If the 8-level transmit FIFO buffer is full, the TX_FULL (SPI_STATUS[3]) bit will be set to 1. When the SPI transmission logic unit draws out the last datum of the transmit FIFO buffer, so that the 8-level transmit FIFO buffer is empty, the TX_EMPTY (SPI_STATUS[2]) bit will be set to 1. Notice that the TX_EMPTY flag is set to 1 while the last transaction is still in progress. In master mode, the software should check both the GO_BUSY (SPI_CTL[0]) bit and TX_EMPTY bit to make sure whether the SPI is in idle or not.

The received FIFO buffer is also an 8-level depth, 32-bit wide, first-in, first-out register buffer. The receive control logic will store the received data to this buffer. The software can read the FIFO buffer data from SPI_RX0 register. There are FIFO related status bits, like RX_EMPTY (SPI_STATUS[0]) and RX_FULL (SPI_STATUS[1]), to indicate the current status of FIFO buffer.

In FIFO mode, the software can set the transmitting and receiving threshold by setting the TX_THRESHOLD (SPI_FFCTL[30:28]) and RX_THRESHOLD (SPI_FFCTL[26:24]) settings. When the count of valid data stored in transmit FIFO buffer is less than or equal to TX_THRESHOLD setting, the TX_INTSTS (SPI_STATUS[10]) bit will be set to 1. When the count of valid data stored in receive FIFO buffer is larger than RX_THRESHOLD setting, the RX_INTSTS (SPI_STATUS[8]) bit will be set to 1.

In FIFO mode, the software can write 8 data to the SPI transmit FIFO buffer in advance. When the SPI controller operates in FIFO mode, the GO_BUSY bit (SPI_CTL[0]) will be controlled by hardware, software should not modify the content of SPI_CTL register unless clearing the FIFOM (SPI_CTL[21]) bit to disable the FIFO mode.

In Master mode transmission operation, the TX_EMPTY flag will be cleared to 0 when the FIFOM bit is set to 1 and the software write the first datum to the SPI_TX0 register. The transmission starts immediately as long as the transmit FIFO buffer is not empty. User can write the next data into SPI_TX0 register immediately. The SPI controller will insert a suspend interval between two successive transactions in FIFO mode and the period of suspend interval is decided by the setting of SP_CYCLE (SPI_CTL [15:12]). User can write data into SPI_TX0 register as long as the TX_FULL flag is 0. Word suspend is only used in FIFO mode.

The subsequent transactions will be triggered automatically if the transmitted data are updated in time. If the SPI_TX0 register does not be updated after all data transfer are done, the transfer will stop.

In Master mode reception operation, the serial data are received from MISOx pin and stored to receive FIFO buffer. The RX_EMPTY flag will be cleared to 0 while the receive FIFO buffer contain unread data. The software can read the received data from SPI_RX0 register as long as the RX_EMPTY flag is 0. If the receive FIFO buffer contains 8 unread data, the RX_FULL flag will be set to 1. The SPI controller will stop receiving data until the software read the SPI_RX0 register.

In Slave mode, when the FIFOM bit is set as 1, the GO_BUSY bit will be set as 1 by hardware automatically. If user wants to stop the slave mode SPI data transfer, both the FIFOM bit and GO_BUSY bit must be cleared to 0 by software.

In Slave mode transmission operation, when the software writes data to SPI_TX0 register, the data will be loaded into transmit FIFO buffer and the TX_EMPTY flag will be set to 0. The transmission will start when the slave device receives clock signal from master. The software can write data to SPI_TX0 register as long as TX_FULL flag is 0. After all data have been drawn out by the SPI transmission logic unit and the software does not update the SPI_TX0 register, the TX_EMPTY flag will be set to 1.

In Slave mode reception operation, the serial data is received from MOSIx pin and stored to SPI_RX0 register. The reception mechanism is similar to master mode receiving operation.

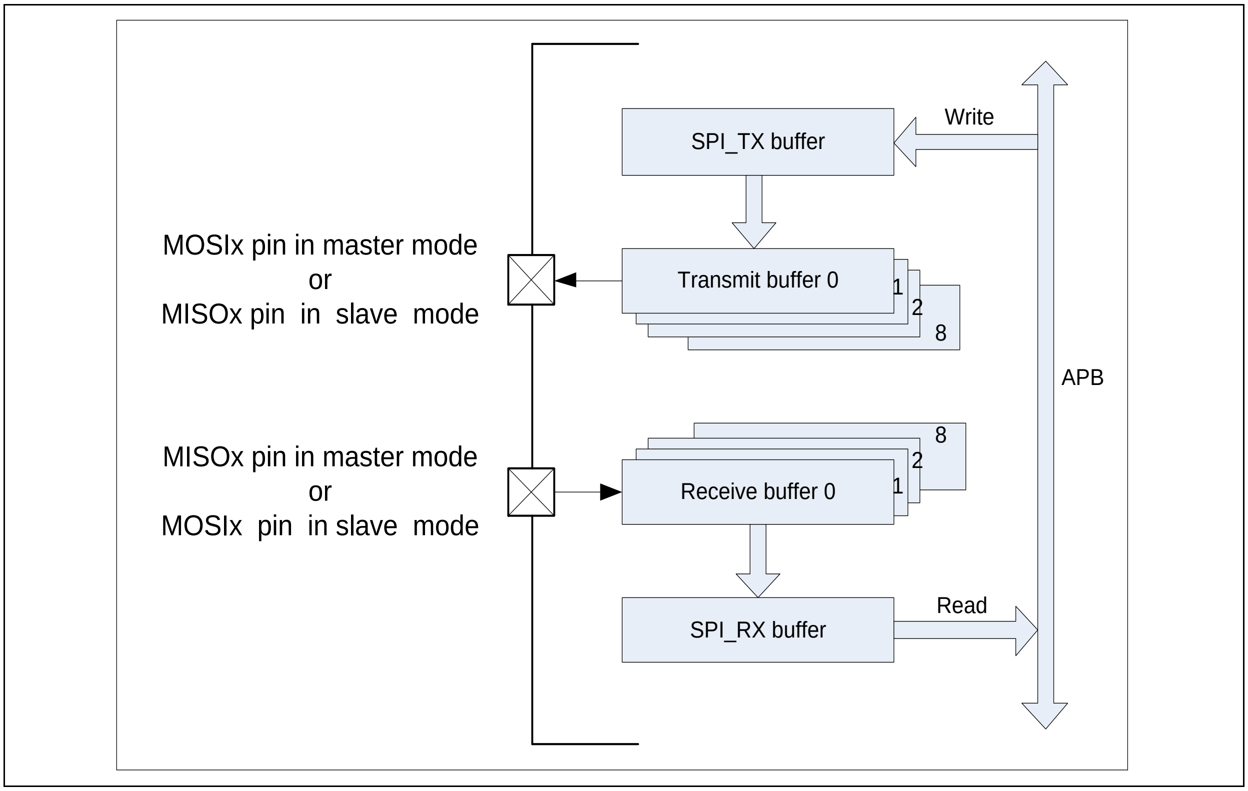


Figure 6-98 SPI FIFO Mode Control

6.17.5.20 Wake-Up

There is wake-up function in the SPI controller. When the system enter Power-down mode by user, it can be waked up by receive a toggle signal of bus clock (SPICLK) from the external device.

6.17.5.21 DMA

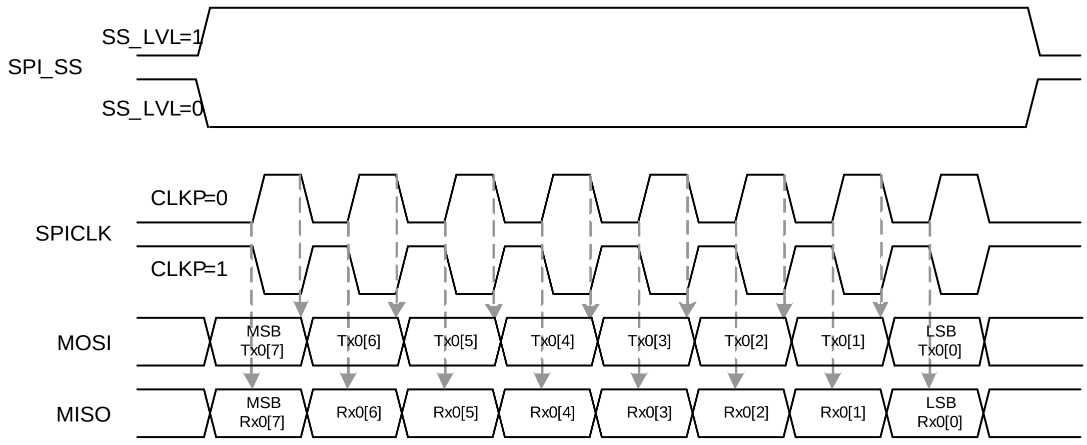
There are transmission and receive DMA function in the SPI controller.

When the TX_DMA_EN (SPI_DMA[0]) bit is set 1, the controller will issue request to PDMA controller to start the DMA transmission process automatically. If using the PDMA mode to transfer data, remember not to set GO_BUSY bit (SPI_CTL[0]). The SPI controller will set it automatically whenever necessary.

When the RX_DMA_EN (SPI_DMA[1]) bit_is set 1, the controller will start the receive PDMA process. SPI controller will issue request to PDMA controller automatically when there is data written into the received buffer or the status of RX_EMPTY (SPI_STATUS[0]) is set to 0 in FIFO mode. If using the receive mode to receive data but the transmission DMA is disable, the GO_BUSY bit shall be set by user.

6.17.5.22 SPI Programming

In Master/Slave mode, the active level of device/slave select (SPI_SS) signal can be programmed to low active or high active in SS_LVL bit (SPI_SSR[2]), but the SPISS0/1 is level trigger or edge trigger which is defined in SS_LTRIG bit (SPI_SSR[4]). The serial clock (SPICLK) idle state can be configured as high state or low state by setting the CLKP bit (SPI_CTL[11]). It also provides the bit length of a transaction in TX_BIT_LEN (SPI_CTL[7:3]), and transmit/receive data from MSB or LSB first in LSB bit (SPI_CTL[10]). Users also can select which edge of serial clock to transmit/receive data in TX_NEG/RX_NEG (SPI_CTL[2:1]) . Four SPI timing diagrams for Master/Slave operations and the related settings are shown below.

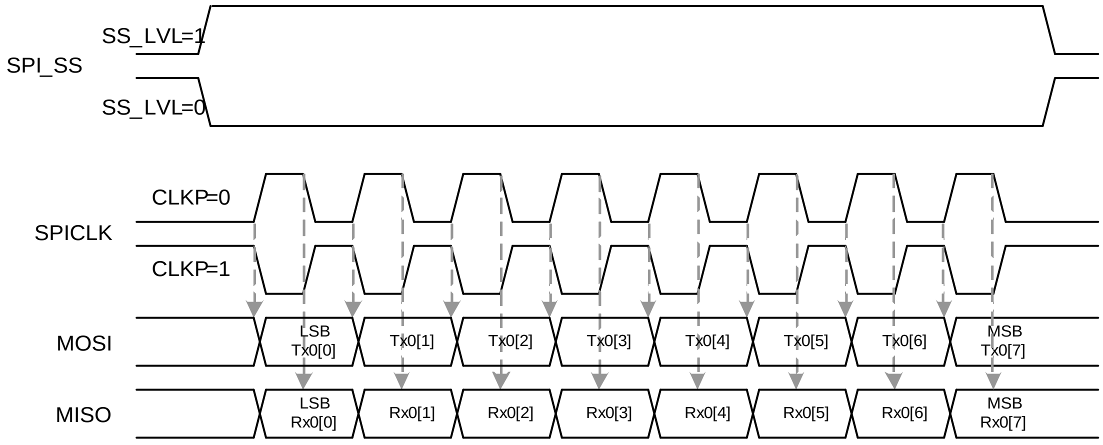


Master Mode: SPI_CTL[SLAVE] =0, SPI_CTL[LSB] =0, SPI_CTL[TX_BIT_LEN] =0x08

1. SPI_CTL[CLKP] =0, SPI_CTL[TX_NEG] =1, SPI_CTL[RX_NEG] =0 or

2. SPI_CTL[CLKP] =1, SPI_CTL[TX_NEG] =0, SPI_CTL[RX_NEG] =1

Figure 6-99 SPI Timing in Master Mode



Master Mode: SPI_CTL[SLAVE] =0, SPI_CTL[LSB] =0, SPI_CTL[TX_BIT_LEN] =0x08

1. SPI_CTL[CLKP] =0, SPI_CTL[TX_NEG] =0, SPI_CTL[RX_NEG] =1 or

2. SPI_CTL[CLKP] =1, SPI_CTL[TX_NEG] =1, SPI_CTL[RX_NEG] =0

Figure 6-100 SPI Timing in Master Mode (Alternate Phase of SPICLK & LSB = 1)

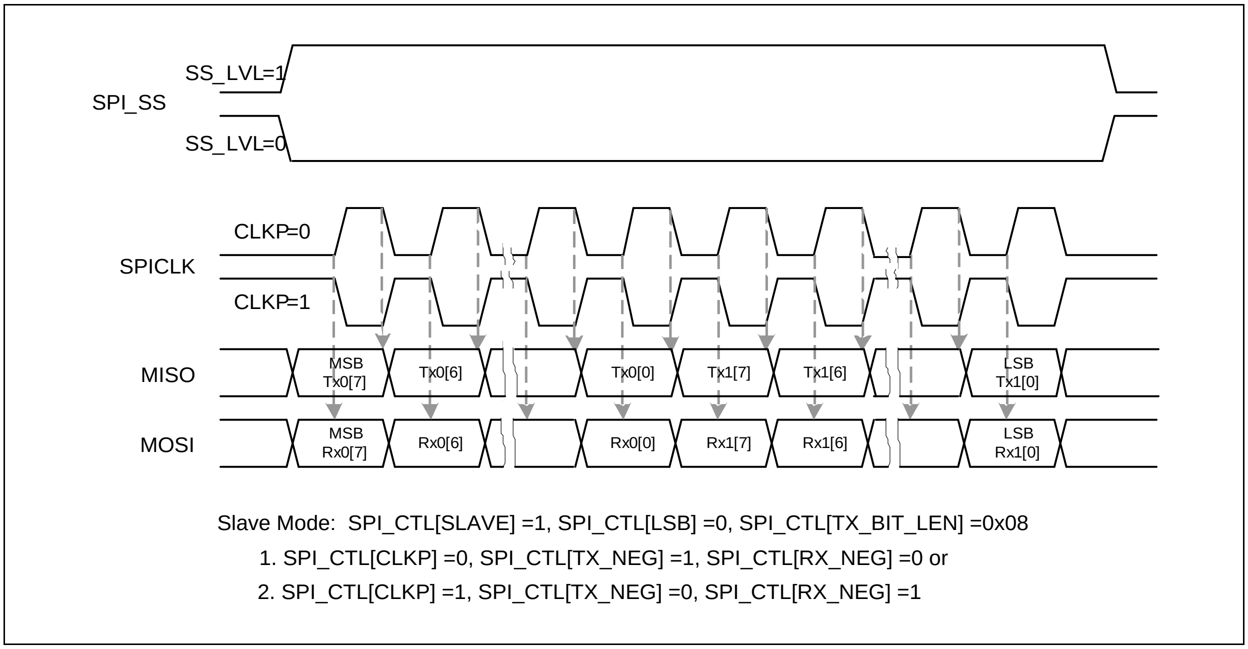

SS_LVL=1
SPI_SS
SS_LVL=0
CLKP=0
SPICLK
CLKP=1
MISO
MSB
Tx0[7]
Tx0[6]
Tx0[0]
Tx1[7]
Tx1[6]
LSB
Tx1[0]
MOSI
MSB
Rx0[7]
Rx0[6]
Rx0[0]
Rx1[7]
Rx1[6]
LSB
Rx1[0]
Slave Mode: SPI_CTL[SLAVE] =1, SPI_CTL[LSB] =0, SPI_CTL[TX_BIT_LEN] =0x08
1. SPI_CTL[CLKP] =0, SPI_CTL[TX_NEG] =1, SPI_CTL[RX_NEG] =0 or
2. SPI_CTL[CLKP] =1, SPI_CTL[TX_NEG] =0, SPI_CTL[RX_NEG] =1

Figure 6-101 SPI Timing in Slave Mode (Alternate Phase of SPICLK & LSB = 0)

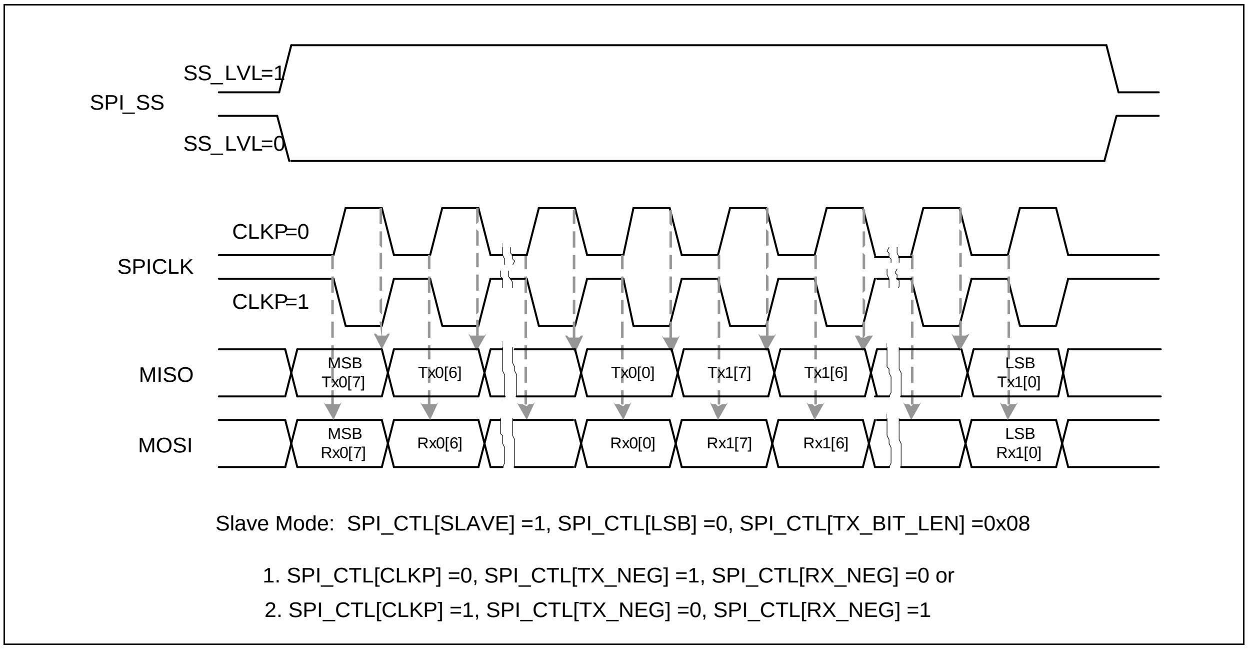
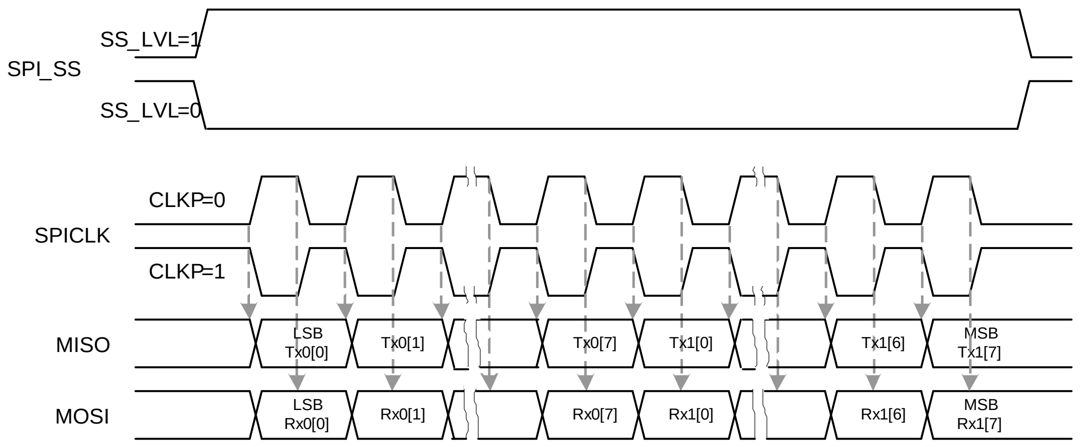

SS_LVL=1
SPI_SS
SS_LVL=0
CLKP=0
SPICLK
CLKP=1
MISO
MSB
Tx0[7]
Tx0[6]
Tx0[0]
Tx1[7]
Tx1[6]
LSB
Tx1[0]
MOSI
MSB
Rx0[7]
Rx0[6]
Rx0[0]
Rx1[7]
Rx1[6]
LSB
Rx1[0]
Slave Mode: SPI_CTL[SLAVE] =1, SPI_CTL[LSB] =0, SPI_CTL[TX_BIT_LEN] =0x08
1. SPI_CTL[CLKP] =0, SPI_CTL[TX_NEG] =1, SPI_CTL[RX_NEG] =0 or
2. SPI_CTL[CLKP] =1, SPI_CTL[TX_NEG] =0, SPI_CTL[RX_NEG] =1

Figure 6-102 SPI Timing in Slave Mode



Slave Mode: SPI_CTL[SLAVE] =1, SPI_CTL[LSB] =1, SPI_CTL[TX_BIT_LEN] =0x08

1. SPI_CTL[CLKP] =0, SPI_CTL[TX_NEG] =0, SPI_CTL[RX_NEG] =1 or

2. SPI_CTL[CLKP] =1, SPI_CTL[TX_NEG] =1, SPI_CTL[RX_NEG] =0

Figure 6-103 SPI Timing in Slave Mode (Alternate Phase of SPICLK)

6.17.5.23 SPI Programming Example

Example 1, SPI controller is set as a master to access an off-chip slave device with following specifications:

- Data bit is received on positive edge of serial clock
- Data bit is transmitted on negative edge of serial clock
- Data bit is transferred from MSB first
- SPICLK is idle at low state
- Only one byte of data to be transmitted/received in a transfer
- Slave select signal is active low

Basically, the specification of the connected off-chip slave device should be referred in details before the following steps:

1. Set the DIVIDER1 (SPI_CLKDIV[7:0]) register to determine the output frequency of serial clock.
2. Write the SPI_SSR register a proper value for the related settings of Master mode:

   (a). to disable the Automatic Slave Select bit AUTOSS (SPI_SSR[3] = 0).

   (b). Select low level trigger output of slave select signal in the Slave Select Active Level bit SS_LVL (SPI_SSR[2] = 0) and Slave Select Level Trigger bit SS_LII(c). Select slave select signal to be output at the I/O pin by setting the respective Slave Select Register bits SSR[0] or SSR[1] (SPI_SSR[1:0]) to active the off-chip slave devices.
3. Write the related settings into the SPI_CTL register

(a). To control this SPI controller as master device in SLAVE bit (SPI_CTL[18] = 0).

(b). Force the serial clock idle state at low in CLKP bit (SPI_CTL[11] = 0).

(c). Select data transmitted at negative edge of serial clock in TX_NEG bit (SPI_CTL[2] = 1).

(d). Select data received at positive edge of serial clock in RX_NEG bit (SPI_CTL[1] = 0).

(e). Set the bit length of transaction as 8 in TX_BIT_LEN bit field (SPI_CTL[7:3] = 0x08).

(f). Set LSB transfer first in LSB bit (SPI_CTL[10] = 1).

4. If this SPI master attempts to transmit (write) one byte data to the off-chip slave device, write the byte data that will be transmitted into the SPI_TX0 register.
5. If this SPI master just only attempts to receive (read) one byte data from the off-chip slave device and does not care what data will be transmitted, the software does not need to update the SPI_TX0 register.
6. Enable the GO_BUSY bit (SPI_CTL[0] = 1) to start the data transfer at the SPI interface.
7. Waiting for SPI interrupt (if the Interrupt Enable INTEN bit(SPI_CTL[17]) is set) or just polling the GO_BUSY bit till it be cleared to 0 by hardware automatically.
8. Read out the received one byte data from RDATA0 (SPI_RX0[7:0]) register.
9. Go to 4) to continue another data transfer or set SSR[0] or SSR[1] to 0 to inactivate the off-chip slave devices.

Example 2, SPI controller is set as a slave device that is controlled by an off-chip master device, and supposes the off-chip master device to access the on-chip SPI slave controller through the SPI interface with the following specifications:

- Data bit is received on positive edge of serial clock
- Data bit is transmitted on negative edge of serial clock
- Data bit is transferred from LSB first
- SPICLK is idle at high state
- Only one byte of data to be transmitted/received in a transfer
- Slave select signal is high level trigger

Basically, the specification of the connected off-chip master device should be referred in details before the following steps

1. Select high level and level trigger for the input of slave select signal in the Slave Select Active Level bit SS_LVL (SPI_SSR[2] = 1) and the Slave Select Level Trigger bit SS_LTRIG (SPI_SSR[4] = 1).
2. Write the related settings into the SPI_CTL register to control this SPI slave actions. Set this SPI controller as slave device in SLAVE bit (SPI_CTL[18] = 1). Select the serial clock idle state at high in CLKP bit (SPI_CTL[11] = 1). Select data transmitted at negative edge of serial clock in TX_NEG bit (SPI_CTL[2] = 1), Select data received at positive edge of serial clock in RX_NEG bit (SPI_CTL[1] = 0). Set the bit length of transaction as 8 bits in TX_BIT_LEN bit field (SPI_CTL[7:3] = 0x08). Set LSB transfer first in LSB bit (SPI_CTL[10] = 1), and don’t care the SP_CYCLE bit field (SPI_CTL[15:12]) due to not burst mode in this case.
3. If this SPI slave attempts to transmit (be read) one byte data to the off-chip master device, write the byte data that will be transmitted into the SPI_TX0 register.
4. If this SPI slave just only attempts to receive (be written) one byte data from the off-chip master device and does not care what data will be transmitted, the software does not need to update the SPI_TX0 register.
5. Enable the GO_BUSY bit (SPI_CTL[0] = 1) to wait for the slave select trigger input and serial clock input from the off-chip master device to start the data transfer at the SPI interface.
6. Waiting for SPI interrupt (if the Interrupt Enable INTEN bit is set) or just polling the GO_BUSY bit till it be cleared to 0 by hardware automatically

7. Read out the received one byte data from (SPI_RX0[7:0])
8. Go to 3) to continue another data transfer or stop data transfer.

6.17.6 Register Map

R: read only, **W:** write only, **R/W:** both read and write, **C:** Write 1 Clear

Register	Offset	R/W	Description	Reset Value
SPI Base Address: **SPIx_BA = 0x4003_0000 + 0x100000*x** **x=0,1**				
SPI_CTL	SPIx_BA+0x00	R/W	SPI Control Register	0x0000_0004
SPI_STATUS	SPIx_BA+0x04	R/W	SPI Status Register	0x0000_0005
SPI_CLKDIV	SPIx_BA+0x08	R/W	SPI Clock Divider Register	0x0000_0000
SPI_SSR	SPIx_BA+0x0C	R/W	SPI Slave Select Register	0x0000_0000
SPI_RX0	SPIx_BA+0x10	R	SPI Receive Data FIFO Register 0	0x0000_0000
SPI_RX1	SPIx_BA+0x14	R	SPI Receive Data FIFO Register 1	0x0000_0000
SPI_TX0	SPIx_BA+0x20	W	SPI Transmit Data FIFO Register 0	0x0000_0000
SPI_TX1	SPIx_BA+0x24	W	SPI Transmit Data FIFO Register 1	0x0000_0000
SPI_VARCLK	SPIx_BA+0x34	R/W	SPI Variable Clock Pattern Flag Register	0x007F_FF87
SPI_DMA	SPIx_BA+0x38	R/W	SPI DMA Control Register	0x0000_0000
SPI_FFCTL	SPIx_BA+0x3C	R/W	SPI FIFO Control Register	0x0000_0000
SPI_INTERNAL	SPIx_BA+0x50	R/W	SPI INTERNAL Register	0x0000_0000

Note: In the following register description, the “x” indicates 0~1 for each SPI module. For example, SPIx_BA includes the sPI0_BAand SPI1_BA.

6.17.7 Register Description

SPI Control Register (SPI_CTL)

Register	Offset	R/W	Description	Reset Value
SPI_CTL	SPIx_BA+0x00	R/W	SPI Control Register	0x0000_0004

31	30	29	28	27	26	25	24
WKEUP_EN	Reserved	DUAL_IO_EN	DUAL_IO_DIR	Reserved	Reserved		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VARCLK_EN	TWOB	FIFOM	Reserved	REORDER	SLAVE	INTEN	Reserved
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SP_CYCLE				CLKP	LSB	Reserved	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TX_BIT_LEN					TX_NEG	RX_NEG	GO_BUSY

Bits	Description	
[31]	**WKEUP_EN**	**Wake-up Enable Control** 0 = Wake-up function Disabled. 1 = Wake-up function Enabled. **Note:** When the system enters Power-down mode, the system can be wake-up from the SPI controller when this bit is enabled and if there is any toggle in the SPICLK port. After the system wake-up, this bit must be cleared by user to disable the wake-up requirement.
[30]	**Reserved**	Reserved.
[29]	**DUAL_IO_EN**	**Dual IO Mode Enable Control** 0 = Dual I/O Mode function Disabled. 1 = Dual I/O Mode function Enabled. Refer to **Dual IO Mode** section.
[28]	**DUAL_IO_DIR**	**Dual IO Mode Direction** 0 = Date read in the Dual I/O Mode function. 1 = Data write in the Dual I/O Mode function.. Refer to **Dual IO Mode** section.
[27:24]	**Reserved**	Reserved.
[23]	**VARCLK_EN**	**Variable Clock Enable Control** 0 = The serial clock output frequency is fixed and only decided by the value of DIVIDER1. 1 = The serial clock output frequency is variable. The output frequency is decided by the value of VARCLK (SPI_VARCLK), DIVIDER1, and DIVIDER2. .Refer to **Variable Clock Function** section.
[22]	**TWOB**	**2-bit Transfer Mode Active** 0 = 2-bit transfer mode Disabled. 1 = 2-bit transfer mode Enabled. Refer to **Two Bit Transfer Mode** section

[21]	**FIFOM**	**FIFO Mode Enable Control** 0 = FIFO mode Disabled (in Normal mode). 1 = FIFO mode Enabled. Refer to **FIFO Mode** section.
[19]	**REORDER**	**Byte Reorder Function Enable Control** 0 = Disable byte reorder function. 1 = Enable byte reorder function and insert a byte suspend interval among each byte. The setting of TX_BIT_LEN must be configured as 00b (32 bits/ word) The suspend interval is defined in SP_CYCLE. Refer to **Byte Reorder** section. **Note:** Byte Suspend is only used in SPI Byte Reorder mode.
[18]	**SLAVE**	**Slave Mode** 0 = SPI controller set as Master mode. 1 = SPI controller set as Slave mode. Refer to **Slave Selection** section
[17]	**INTEN**	**Interrupt Enable Control** 0 = SPI Interrupt Disabled. 1 = SPI Interrupt Enabled.
[16]	**Reserved**	Reserved.
[15:12]	**SP_CYCLE**	**Suspend Interval (Master Only)** These four bits provide configurable suspend interval between two successive transmit/receive transaction in a transfer. The suspend interval is from the last falling clock edge of the current transaction to the first rising clock edge of the successive transaction if CLKP = “0”. If CLKP = “1”, the interval is from the rising clock edge to the falling clock edge. The default value is 0x3. The desired suspend interval is obtained according to the following equation:)(SP_]YCLE[3]0) + 0.5) * period of SPICLK Ex: SP_CYCLE = 0x0 …. 0.5 SPICLK clock cycle. SP_CYCLE = 0x1 …. 1.5 SPICLK clock cycle. …… SP_CYCLE = 0xE …. 14.5 SPICLK clock cycle. SP_CYCLE = 0xF …. 15.5 SPICLK clock cycle. If the Variable Clock function is enabled, the minimum period of suspend interval (the transmit data in FIFO buffer is not empty) between the successive transaction is (6.5 + SP_CYCLE) * SPICLK clock cycle.
[11]	**CLKP**	**Clock Polarity** 0 = The default level of SCLK is low. 1 = The default level of SCLK is high. Refer to **Clock Parity** section.
[10]	**LSB**	**Send LSB First** 0 = The MSB, which bit of transmit/receive register depends on the setting of TX_BITLEN, is transmitted/received first. 1 = The LSB, bit 0 of the SPI_TX0/1, is sent first to the the SPI data output pin, and the first bit received from the SPI data input pin will be put in the LSB position of the SPI_RX register (SPI_RX0/1). Refer to **LSB first** section.

[9:8]		Reserved.
[7:3]	**TX_BIT_LEN**	**Transmit Bit Length** This field specifies how many bits can be transmitted / received in one transaction. The minimum bit length is 8 bits and can be up to 32 bits. 01000 = 8 bits are transmitted in one transaction. 01001 = 9 bits are transmitted in one transaction. 01010 = 10 bits are transmitted in one tran. ----- 11111 = 31 bits are transmitted in one transaction. 00000 = 32 bits are transmitted in one transaction.
[2]	**TX_NEG**	**Transmit At Negative Edge** 0 = The transmitted data output is changed on the rising edge of SPI_SCLK. 1 = The transmitted data output is changed on the falling edge of SPI_SCLK. Refer to **Edge** section.
[1]	**RX_NEG**	**Receive At Negative Edge** 0 = The received data is latched on the rising edge of SPI_SCLK. 1 = The received data is latched on the falling edge of SPI_SCLK. Refer to **Edge** section.
[0]	**GO_BUSY**	**SPI Transfer Control Bit and Busy Status** 0 = Writing this bit “0” will stop data transfer if SPI is transferring. 1 = In Master mode, writing “1” to this bit will start the SPI data transfer; In Slave mode, writing ‘1’ to this bit indicates that the salve is ready to communicate with a master. If the FIFO mode is disabled, during the data transfer, this bit keeps the value of ‘1’. As the transfer is finished, this bit will be cleared automatically. Software can read this bit to check if the SPI is in busy status. In FIFO mode, this bit will be controlled by hardware. Software should not modify this bit. In slave mode, this bit always returns 1 when software reads this register. In master mode, this bit reflects the busy or idle status of SPI. **Note:** 1.When FIFO mode is disabled, all configurations should be set before writing “1” to the GO_BUSY bit in the SPI_CTL register. 2. When FIFO bit is disabled and the software uses TX or RX PDMA function to transfer data, this bit will be cleared after the PDMA controller finishes the data transfer.

SPI Status Register (SPI_STATUS)

Register	Offset	R/W	Description	Reset Value
SPI_STATUS	SPIx_BA+0x04	R/W	SPI Status Register	0x0000_0005

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TX_FIFO_CNT				**RX_FIFO_CNT**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**TIME_OUT_STS**	**Reserved**	**TXINT_STS**	**RX_OVER_RUN**	**RXINT_STS**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTSTS	**SLV_START_INTSTS**	**Reserved**	**LTRIG_FLAG**	**TX_FULL**	**TX_EMPTY**	**RX_FULL**	**RX_EMPTY**

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:20]	**TX_FIFO_CNT**	**Data counts in TX FIFO (Read Only)**
[19:16]	**RX_FIFO_CNT**	**Data counts in RX FIFO (Read Only)**
[15:13]	**Reserved**	Reserved.
[12]	**TIME_OUT_STS**	**TIMEOUT Interrupt Flag** 0 = There is not timeout event on the received buffer. 1 = Time out event active in RX FIFO is not empty. Refer to **Time Out** section. **Note:** This bit will be cleared by writing 1 to it.
[11]	**Reserved**	Reserved.
[10]	**TXINT_STS**	**TX FIFO Threshold Interrupt Status (Read Only)** 0 = TX valid data counts bigger than **TXTHRESHOLD (SPI_FFCTL[31:28].** 1 = TX valid data counts small or equal than **TXTHRESHOLD.**
[9]	**RX_OVER_RUN**	**RX FIFO Over Run Status** 0 = No FIFO is over run. 1 = Receive FIFO over run. **Note 1:** If SPI receives data when RX FIFO is full, this bit will set to 1, and the received data will dropped. **Note 2:** This bit will be cleared by writing 1 to it.
[8]	**RXINT_STS**	**RX FIFO Threshold Interrupt Status (Read Only)** 0 = RX valid data counts small or equal than **RXTHRESHOLD (SPI_FFCTL[27:24]).** 1 = RX valid data counts bigger than **RXTHRESHOLD.** **Note:** If RXINT_EN(SPI_FFCTL[2]) = 1 and RX_INTSTS = 1, SPI will generate interrupt.
[7]	**INTSTS**	**Interrupt Status**

		0 = Transfer is not finished yet. 1 = Transfer is done. The interrupt is requested when the INTEN(SPI_CTL[17]) bit is enabled. **Note:** This bit is read only, but can be cleared by writing “1” to this bit.
[6]	**SLV_START_INTSTS**	**Slave Start Interrupt Status** It is used to dedicate that the transfer has started in Slave mode with no slave select. 0 = Slave started transfer no active. 1 = Transfer has started in Slave mode with no slave select. It is auto clear by transfer done or writing one clear.
[4]	**LTRIG_FLAG**	**Level Trigger Accomplish Flag (INTERNAL ONLY)** In Slave mode, this bit indicates whether the received bit number meets the requirement or not after the current transaction done. 0 = The transferred bit length of one transaction does not meet the specified requirement. 1 = The transferred bit length meets the specified requirement which defined in TX_BIT_LEN. **Note:** This bit is READ only. As the software sets the GO_BUSY bit to 1, the LTRIG_FLAG will be cleared to 0 after 4 SPI engine clock periods plus 1 system clock period. In FIFO mode, this bit is unmeaning.
[3]	**TX_FULL**	**Transmitted FIFO_FULL Status** 0 = Transmitted data FIFO is not full in the FIFO mode. 1 = Transmitted data FIFO is full in the FIFO mode.
[2]	**TX_EMPTY**	**Transmitted FIFO_EMPTY Status** 0 = Transmitted data FIFO is not empty in the FIFO mode. 1 =Transmitted data FIFO is empty in the FIFO mode.
[1]	**RX_FULL**	**Received FIFO_FULL Status** 0 = Received data FIFO is not full in FIFO mode. 1 = Received data FIFO is full in the FIFO mode.
[0]	**RX_EMPTY**	**Received FIFO_EMPTY Status** 0 = Received data FIFO is not empty in the FIFO mode. 1 = Received data FIFO is empty in the FIFO mode.

SPI Serial Clock Divider Register (SPI_CLKDIV)

Register	Offset	R/W	Description	Reset Value
SPI_CLKDIV	SPIx_BA+0x08	R/W	SPI Clock Divider Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DIVIDER2							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DIVIDER1							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**DIVIDER2**	**Clock Divider 2** The value is the 2nd frequency divider of the PCLK to generate the serial clock of SPI_SCLK. The desired frequency is obtained according to the following equation: $f_{sclk} = \frac{f_{eclk}}{(DIVIDER2+1)*2}$
[15:8]	**Reserved**	Reserved.
[7:0]	**DIVIDER1**	**Clock Divider 1** The value is the 1th frequency divider of the PCLK to generate the serial clock of SPI_SCLK. The desired frequency is obtained according to the following equation: $f_{sclk} = \frac{f_{eclk}}{(DIVIDER1+1)}$ Where f_{eclk} is the SPI peripheral clock source. It is defined in the CLK_SEL2[21:20] in Clock control section (CLK_BA + 0x18).

SPI Slave Select Register (SPI_SSR)

Register	Offset	R/W	Description	Reset Value
SPI_SSR	SPIx_BA+0x0C	R/W	SPI Slave Select Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**SS_INT_OPT**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**SSTA_INTEN**	**SLV_ABORT**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**NOSLVSEL**	**SS_LTRIG**	**AUTOSS**	**SS_LVL**	**SSR**	

Bits	Description	
[31:10]	**Reserved**	Reserved.
[16]	**SS_INT_OPT**	**Slave Select Interrupt Option** It is used to enable the interrupt when the transfer has done in slave mode. 0 = No any interrupt, even there is slave select inactive event. 1 = There is interrupt event when the slave select becomes inactive from active condition. It is used to inform the user to know that the transaction has finished and the slave select into the inactive state.
[9]	**SSTA_INTEN**	**Slave Start Interrupt Enable Control** 0 = Transfer start interrupt Disabled in no slave select mode. 1 = Transaction start interrupt Enabled in no slave select mode. It is cleared when the current transfer done or the SLV_START_INTSTS bit cleared (write 1 clear). Refer to **No Slave Select Mode**.
[8]	**SLV_ABORT**	**Abort in Slave Mode with No Slave Selected** 0 = No force the slave abort. 1 = Force the current transfer done in no slave select mode. Refer to **No Slave Select Mode**. **Note:** It is auto cleared to “0” by hardware when the abort event is active.
[5]	**NOSLVSEL**	**No Slave Selected in Slave Mode** This is used to ignore the slave select signal in Slave mode. The SPI controller can work on 3 wire interface including SPICLK, SPI_MISO, and SPI_MOSI when it is set as a slave device. 0 = The controller is 4-wire bi-direction interface. 1 = The controller is 3-wire bi-direction interface in Slave mode. When this bit is set as 1, the controller start to transmit/receive data after the GO_BUSY bit active and the serial clock input. Refer to **No Slave Select Mode**. **Note:** In no slave select signal mode, the SS_LTRIG (SPI_SSR[4]) shall be set as “1”.
[4]	**SS_LTRIG**	**Slave Select Level Trigger**

		0 = The input slave select signal is edge-trigger. 1 = The slave select signal will be level-trigger. It depends on SS_LVL to decide the signal is active low or active high.
[3]	**AUTOSS**	**Automatic Slave Selection (Master Only)** 0 = If this bit is set as "0", slave select signals are asserted and de-asserted by setting and clearing related bits in SSR[1:0] register. 1 = If this bit is set as "1", SPISS[1:0] signals are generated automatically. It means that device/slave select signal, which is set in SSR[1:0] register is asserted by the SPI controller when transmit/receive is started, and is de-asserted after each transaction is done.
[2]	**SS_LVL**	**Slave Select Active Level** It defines the active level of device/slave select signal (SPISS[1:0]). 0 = The SPI_SS slave select signal is active Low. 1 = The SPI_SS slave select signal is active High.
[1:0]	**SSR**	**Slave Select Active Register (Master Only)** If AUTOSS bit (SPI_SSR[3]) is cleared, writing "1" to SSR[0] bit sets the SPISS[0] line to an active state and writing "0" sets the line back to inactive state.(the same as SSR[1] for SPISS[1]) AUTOSS = 0. 00 = Both SPISS[1] and SPISS[0] are inactive. 01 = SPISS[1] is inactive, SPISS[0] is active. 10 = SPISS[1] is active, SPISS[0] is inactive. 11 = Bothe SPISS[1] and SPISS[0] are active.. If AUTOSS bit is set, writing "1" to any bit location of this field will select appropriate SPISS[1:0] line to be automatically driven to active state for the duration of the transaction, and will be driven to inactive state for the rest of the time. (The active level of SPISS[1:0] is specified in SS_LVL). AUTOSS =1. 00 = Both SPISS[1] and SPISS[0] are inactive. 01 = SPISS[1] is inactive, SPISS[0] is active on the duration of transaction. 10 = SPISS[1] is active on the duration of transaction, SPISS[0] is inactive. 11 = Bothe SPISS[1] and SPISS[0] are active on the duration of transaction. **Note:** 1. This interface can only drive one device/slave at a given time. Therefore, the slaves select of the selected device must be set to its active level before starting any read or write transfer. 2. SPISS[0] is also defined as device/slave select input in Slave mode. And that the slave select input must be driven by edge active trigger which level depend on the SS_LVL setting, otherwise the SPI slave core will go into dead path until the edge active triggers again or reset the SPI core by software.

SPI Receive FIFO Register (SPI_RX)

Register	Offset	R/W	Description	Reset Value
SPI_RX0	SPIx_BA+0x10	R	SPI Receive Data FIFO Register 0	0x0000_0000
SPI_RX1	SPIx_BA+0x14	R	SPI Receive Data FIFO Register 1	0x0000_0000

31	30	29	28	27	26	25	24
RXDATA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RXDATA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RXDATA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RXDATA							

Bits	Description	
[31:0]	**RXDATA**	**Receive Data FIFO Bits(Read Only)** The received data can be read on it. If the FIFO bit is set as 1, the user also checks the RX_EMPTY, SPI_STATUS[0], to check if there is any more received data or not. **Note:** 1. The SPI_RX1 is used only in TWOB bit (SPI_CTL[22]) is set 1. The first channel's received data shall be read from SPI_RX0 and the second channel's received data shall be read from SPI_RX1 in two-bit mode. SPI_RX0 shall be read first in TWOB mode. In FIFO and two-bit mode, the first read back data in SPI_RX0 is the first channel data and the second read back data in SPI_RX0 is the second channel data. 2. These registers are read only.

SPI Transmit Data FIFO Register (SPI_TX)

Register	Offset	R/W	Description	Reset Value
SPI_TX0	SPIx_BA+0x20	W	SPI Transmit Data FIFO Register 0	0x0000_0000
SPI_TX1	SPIx_BA+0x24	W	SPI Transmit Data FIFO Register 1	0x0000_0000

31	30	29	28	27	26	25	24
TDATA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TDATA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TDATA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TDATA							

Bits	Description	
[31:0]	**TDATA**	**Transmit Data FIFO Bits(Write Only)** The Data Transmit Registers hold the data to be transmitted in the next transfer. The number of valid bits depends on the setting of transmit bit length field of the SPI_CTL register. For example, if TX_BIT_LEN is set to 0x8, the bit SPI_TX[7:0] will be transmitted in next transfer. If TX_BIT_LEN is set to 0x0, the SPI controller will perform a 32-bit transfer. **Note:** 1. The SPI_TX1 is used only in TWOB bit (SPI_CTL[22]) is set 1. The first channel's transmitted data shall be written into SPI_TX0 and the second channel's transmitted data shall be written into SPI_TX1 in two-bit mode. SPI_TX0 shall be written first in TWOB mode. In FIFO and two-bit mode, the first written into data in SPI_TX0 is the first channel's transmitted data and the second written data in SPI_RX0 is the second channel's transmitted data. 2. When the SPI controller is configured as a slave device and the FIFO mode is disabled, if the SPI controller attempts to transmit data to a master, the software must update the transmit data register before setting the GO_BUSY bit to 1.

SPI Variable Clock Register (SPI_VARCLK)

Register	Offset	R/W	Description	Reset Value
SPI_VARCLK	SPIx_BA+0x34	R/W	SPI Variable Clock Pattern Flag Register	0x007F_FF87

31	30	29	28	27	26	25	24
VARCLK							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VARCLK							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VARCLK							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VARCLK							

Bits	Description	
[31:0]	**VARCLK**	**Variable Clock Pattern Flag** The value in this field is the frequency patterns of the SPICLK. Refer to **Variable Clock Function** section for detail information. **Note:** It is used for CLKP = 0 only.

SPI DMA Control Register (SPI_DMA)

Register	Offset	R/W	Description	Reset Value
SPI_DMA	SPIx_BA+0x38	R/W	SPI DMA Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					PDMA_RST	RX_DMA_EN	TX_DMA_EN

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**PDMA_RST**	**PDMA Reset** It is used to reset the SPI PDMA function into default state. 0 = After reset PDMA function or in normal operation. 1 = Reset PDMA function. **Note:** it is auto cleared to “0” after the reset function has done.
[1]	**RX_DMA_EN**	**Receiving PDMA Enable Control** 0 = Receiver PDMA function Disabled. 1 = Receiver PDMA function Enabled. Refer to **DMA** section for more detail information. **Note:** Hardware will clear this bit to 0 automatically after PDMA transfer done. In Slave mode and the FIFO bit is disabled, if the receive PDMA is enabled but the transmit PDMA is disabled, the minimal suspend interval between two successive transactions input is need to be larger than 9 SPI peripheral clock + 4 APB clock for edge mode and 9.5 SPI peripheral clock + 4 APB clock.
[0]	**TX_DMA_EN**	**Transmit PDMA Enable Control** 0 = Transmit PDMA function Disabled. 1 = Transmit PDMA function Enabled. Refer to **DMA** section for more detail information. SPI_CTL**Note:** 1. Two transaction need minimal 18 APB clock + 8 SPI peripheral clocks suspend interval in master mode for edge mode and 18 APB clock + 9.5 SPI peripheral clocks for level mode. 2. If the 2-bit function is enabled, the requirement timing shall append 18 APB clock based on the above clock period. Hardware will clear this bit to 0 automatically after PDMA transfer done.

SPI FIFO Control Register (SPI_FFCTL)

Register	Offset	R/W	Description	Reset Value
SPI_FFCTL	SPIx_BA+0x3C	R/W	SPI FIFO Control Register	0x0000_0000

31	30	29	28	27	26	25	24
	TX_THRESHOLD				**RX_THRESHOLD**		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TIMEOUT_EN	**Reserved**		**RXOVE_INTEN**	**TXINT_EN**	**RX_INTEN**	**TX_CLR**	**RX_CLR**

Bits	Description	
[31]	**Reserved**	Reserved.
[30:28]	**TX_THRESHOLD**	**Transmit FIFO Threshold** If TX valid data counts small or equal than TXTHRESHOLD, TXINT_STS (SPI_STATUS[10]) will set to 1.
[26:24]	**RX_THRESHOLD**	**Received FIFO Threshold** If RX valid data counts large than RXTHRESHOLD, RXINT_STS (SPI_STATUS[8]) will set to 1,.
[23:8]	**Reserved**	Reserved.
[7]	**TIMEOUT_EN**	**RX Read Time Out Function Enable Control** 0 = RX read Timeout function Disabled. 1 = RX read Timeout function Enable.
[6:5]	**Reserved**	Reserved.
[4]	**RXOVINT_EN**	**RX FIFO Over Run Interrupt Enable Control** 0 = RX FIFO over run interrupt Disabled. 1 = RX FIFO over run interrupt Enable.
[3]	**TXINT_EN**	**TX Threshold Interrupt Enable Control** 0 = TX threshold interrupt Disabled. 1 = TX threshold interrupt Enable.
[2]	**RXINT_EN**	**RX Threshold Interrupt Enable Control** 0 = Rx threshold interrupt Disabled. 1 = RX threshold interrupt Enable.
[1]	**TX_CLR**	**Transmitting FIFO Counter Clear** 0 = No clear the transmitted FIFO. 1 = Clear the transmitted FIFO. **Note:** This bit is used to clear the transmit counter in FIFO Mode. This bit can be written “1” to clear the transmitting counter and this bit will be cleared to “0” automatically after

		clearing transmitting counter. After the clear operation, the flag of TX_EMPTY in SPI_STATUS[2] will be set to “1”.
[0]	**RX_CLR**	**Receiving FIFO Counter Clear** 0 = No clear the received FIFO. 1 = Clear the received FIFO. **Note:** This bit is used to clear the receiver counter in FIFO Mode. This bit can be written “1” to clear the receiver counter and this bit will be cleared to “0” automatically after clearing receiving counter. After the clear operation, the flag of RX_EMPTY in SPI_STATUS[0] will be set to “1”.

6.18 LCD Display Driver

6.18.1 Overview

The LCD driver can directly drive a LCD glass by creating the ac segment and common voltage signals automatically. It can support static, 1/2 duty, 1/3 duty, 1/4 duty, 1/5 duty and 1/6 duty LCD glass with up to 3 segments with 6 COM (segment 0 is used as LCD_COM4 and segment 1 is used as LCD_COM5) or 36 segments with 4 COM (LCD_COM0 ~ LCD_COM3).

A built-in charge pump function can be enabled to provide the LCD glass with higher voltage than the system voltage. The LCD driver would generate voltage higher than the threshold voltage in older to darken a segment and a voltage lower than threshold to make a segment clear. However, the LCD display segment will degrade if the applied voltage has a DC-component. To avoid this, the generated waveform by LCD driver are arranged such that average voltage of each segment is 0 and the RMS(root-mean-square) voltage applied on a LCD segment lower than the segment threshold making LCD clear and RMS voltage higher than the segment threshold making LCD dark.

6.18.2 Features

- Supports Segment/Com:
  - 108 dots (6x18) or 80 dots (4x20) in LQFP48 package
  - 108 dots (6x18) or 80 dots (4x20) or 132 dots (6x 22) or 96 dots (4x24) or 180 dots (6x30) or 128 dots (4x32) in LQFP64 package
  - 204 dots (6x34) or 144 dots (4x36) in LQFP100 package
- Common 0-5 multiplexing functions with GPI/O pins
- Segment 0-35 multiplexing function with GPI/O pins
- Supports Static,1/2 bias and 1/3 bias voltage
- Six display modes: Static,1/2 duty, 1/3 duty, 1/4 duty, 1/5 duty or 1/6 duty Selectable LCD frequency by frequency divider
- Configurable frame frequency
- Internal Charge pump, adjustable contrast adjustment
- Embedded LCD bias reference ladder (R-Type, 200/300/400 kΩ resisters)
- Configurable Charge pump frequency
- Blinking capability
- Supports R/C/Ext_C-type method
- LCD frame interrupt

6.18.3 Block Diagram

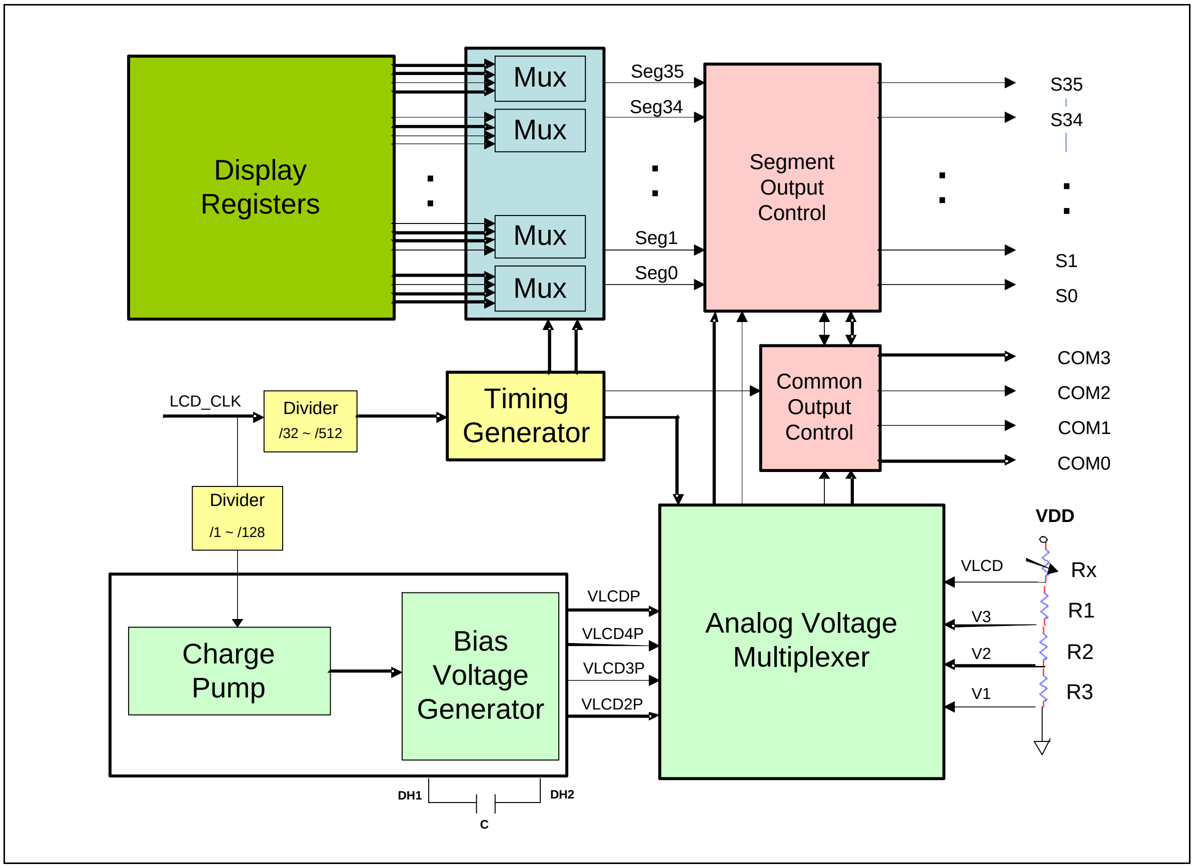


Figure 6-104 LCD Driver Block Diagram

6.18.4 Functional Description

The LCD driver consists of display memory register, segment output control, common output, timing generator, charge pump and analog voltage multiplexer blocks. The display memory register stores LCD segment darkened or cleared data. The data bit that is stored in display memory register with “1” makes the LCD segment be darkened and the data bit that is stored in display memory register with “0” makes the LCD segment be cleared. The display memory register is organized with LCD_MEM_0 ~ LCD_MEM_8 registers. Programming the data bits in LCD_MEM_0 ~ LCD_MEM_8 registers can make the corresponding LCD segment be darkened or cleared. The data stored in display memory register is multiplexed to segment output block sequentially with clock generated by timing generator block. The segment output block is in charge of producing SEG 0 ~ SEG 35 driving line and the common output block is in charge of producing COM0 ~ COM3 driving line. Charge pump block provides boosting voltage function for LCD glass. The charge pump input voltage range is from 1.8V to 3.6V. The multi-levels bias voltage can be programmed by CPUMP_VOL_SET (LCD_DISPCTL[10:8]) and the multi-levels bias voltage from 2.6V to 3.3 V can be generated by charge pump block. The analog voltage multiplexer can generates static, 1/2 bias and 1/3 bias voltage output by setting BIAS_SEL (LCD_DISPCTL[2:1]). User can program the BIAS_SEL bits to generate different bias voltage for COM and SEG driving line to drive LCD glass. Each common signal is selected sequentially according to the specified number of time slices of its frame period. For example, in 1/3 duty, COM0 to COM2 will output waveforms, COM3 will be tied to low. Whereas for 1/6 duty, COM0 to COM5 will output waveforms. The COM signal waveform is shown in Figure 6-106.

6.18.4.1 LCD Display Memory MAP

DISPLAY Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COM	X	X	COM5	COM4	COM3	COM2	OCM1	COM0	X	X	COM5	COM4	COM3	COM2	OCM1	COM0	X	X	COM5	COM4	COM3	COM2	OCM1	COM0	X	X	COM5	COM4	COM3	COM2	OCM1	COM0
LCD_MEM_8	X	X	SEG35	SEG35	SEG35	SEG35	SEG35	SEG35	X	X	SEG34	SEG34	SEG34	SEG34	SEG34	SEG34	X	X	SEG33	SEG33	SEG33	SEG33	SEG33	SEG33	X	X	SEG32	SEG32	SEG32	SEG32	SEG32	SEG32
LCD_MEM_7	X	X	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	X	X	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	X	X	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	X	X	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28
LCD_MEM_6	X	X	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	X	X	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	X	X	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	X	X	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24
LCD_MEM_5	X	X	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	X	X	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	X	X	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	X	X	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20
LCD_MEM_4	X	X	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	X	X	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	X	X	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	X	X	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16
LCD_MEM_3	X	X	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	X	X	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	X	X	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	X	X	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12
LCD_MEM_2	X	X	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	X	X	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	X	X	SEG09	SEG09	SEG09	SEG09	SEG09	SEG09	X	X	SEG08	SEG08	SEG08	SEG08	SEG08	SEG08
LCD_MEM_1	X	X	SEG07	SEG07	SEG07	SEG07	SEG07	SEG07	X	X	SEG06	SEG06	SEG06	SEG06	SEG06	SEG06	X	X	SEG05	SEG05	SEG05	SEG05	SEG05	SEG05	X	X	SEG04	SEG04	SEG04	SEG04	SEG04	SEG04
LCD_MEM_0	X	X	SEG03	SEG03	SEG03	SEG03	SEG03	SEG03	X	X	SEG02	SEG02	SEG02	SEG02	SEG02	SEG02	X	X	SEG01	SEG01	SEG01	SEG01	SEG01	SEG01	X	X	SEG00	SEG00	SEG00	SEG00	SEG00	SEG00

Figure 6‑105 LCD Memory Map

6.18.4.2 Frame Counter (FC) and Blinking Display

In 6-mux configuration, COM0, COM1, COM2, COM3, COM4 and COM5 organize one frame. In 5-mux configuration, COM0, COM1, COM2, COM3 and COM4 organize one frame. In 4-mux configuration, COM0, COM1, COM2 and COM3 organize one frame. In 3-mux configuration, COM0, COM1 and COM2 organize one frame. In 2-mux configuration, COM0 and COM1 organize one frame. In static configuration, COM0 organizes one frame. The frame counter can be pre-scaled by programming pre-scale counter (PRESCL (LCD_FCR[3:2])). The pre-scale counter can be divided by 1, 2, 4 and 8. The frame counter is counted down from FCV (LCD_FCR[9:4]) to 0. FCV is the top value of frame counter. If FCEN (LCD_FCR[0]) is set to 1 and FCINTEN (LCD_FCR[1]) is set to 1, once frame counter is counted down to 0, the frame counter overflow interrupt is generated. At the same time, the frame counter is reloaded with FCV automatically. The LCD blinking display is controlled with frame counter overflow time. In blinking configuration, the segments are turned on and turned off alternately by frame counter overflow time. The frame counter overflowing interrupt can be also used as synchronization for filling data to LCD display memory register.

6.18.4.3 LCD Display Power Down

If the power-down request is triggered from system manager, LCD controller will execute the frame completely to avoid the DC component. When the frame is executed completely, the LCD power down interrupt signal is generated to inform system manager the LCD controller is ready to enter power down state, if PDINT_EN (LCD_CTL[9]) is enabled. Otherwise, if PDINT_EN (LCD_CTL[9]) is disabled, the LCD power down interrupt signal is blocked and the interrupt is disabled. If the PDDSIP_EN (LCD_CTL[8]) is set to 1, the LCD display is operated in Power-down mode. Otherwise, if PDDSIP_EN (LCD_CTL[8]) is cleared to 0, the LCD display is off in Power-down mode.

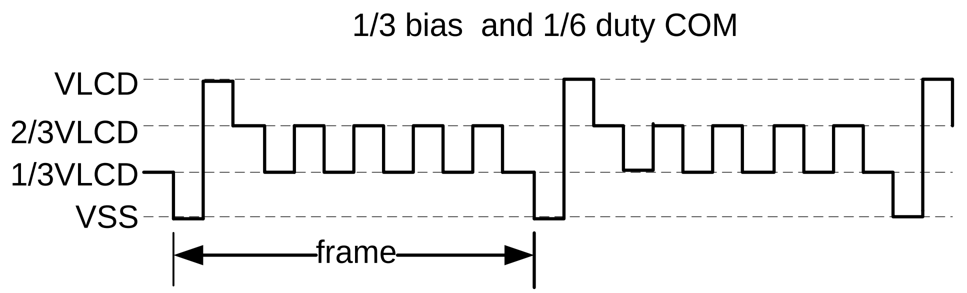


Figure 6‑106 COM Signal Waveform

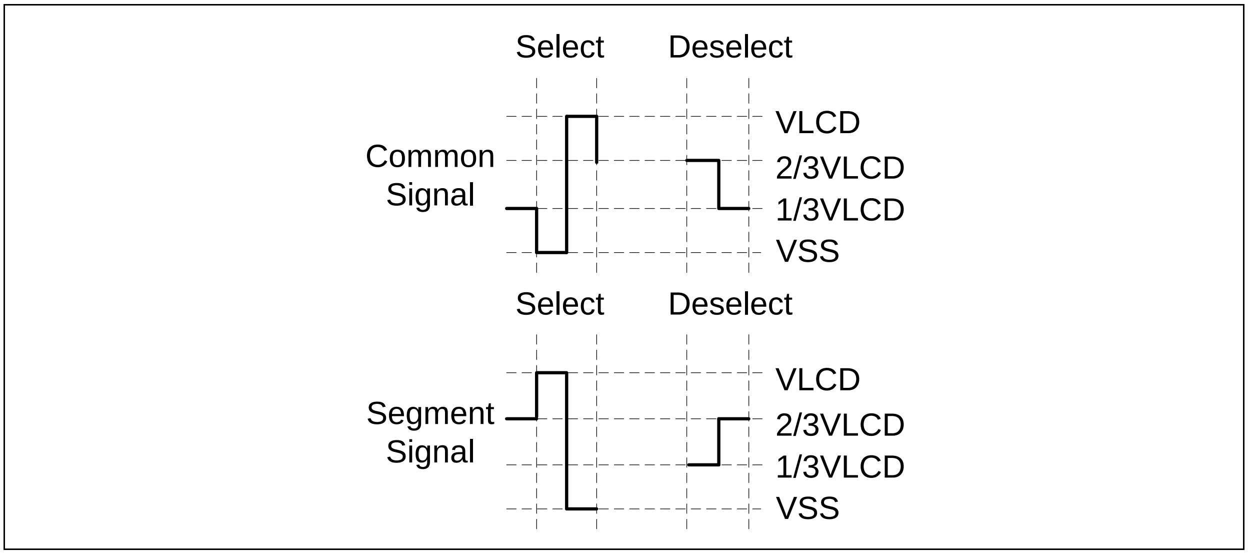


Figure 6‑107 SEG Signal Waveform

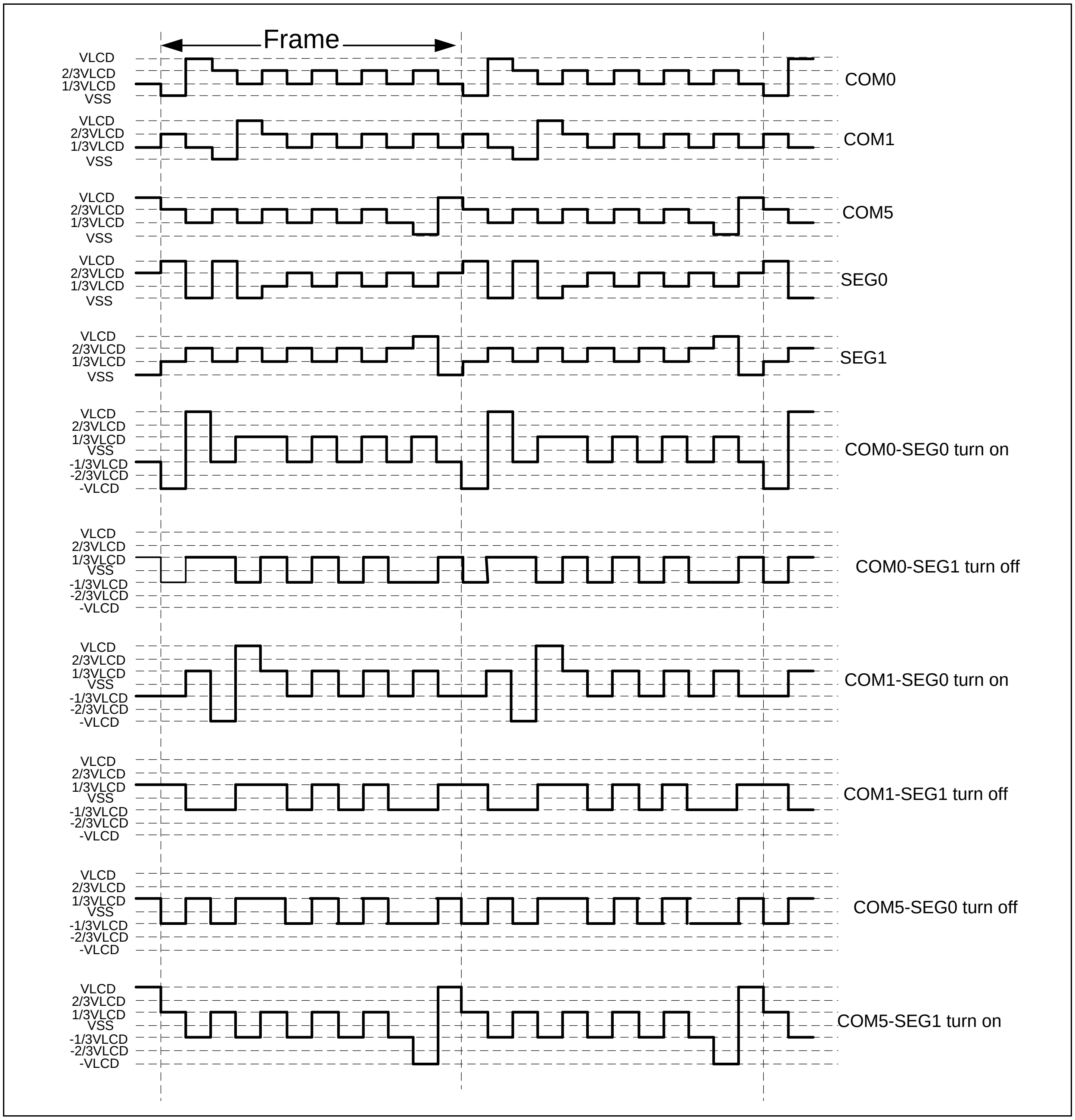


Figure 6-108 COM-SEG Signal Waveform by 1/6 Duty with 1/3 Bias

6.18.4.4 LCD setting for different types

	R-Type (External Resister Ladder)	R-Type (Internal Resister Ladder)	C-Type	Ext_C-Type
CPUMP_EN (LCD_DISPCTL[0])	0	0	1	1
BIAS_SEL (LCD_DISPCTL[2:1])	0 ~ 3	0 ~ 3	0 ~ 3	0 ~ 3
IBRL_EN (LCD_DISPCTL[4])	0	1	0	0
BV_SEL (LCD_DISPCTL[6])	1	1	0	0
CPUMP_VOL_SET (LCD_DISPCTL[10:8])	-	-	0 ~ 7	0 ~ 7
CPUMP_FREQ (LCD_DISPCTL[13:11])	-	-	0 ~ 7	0 ~ 7
Ext_C (LCD_DISPCTL[16])	0	0	0	1[1]
Res_Sel (LCD_DISPCTL[18:17])	-	set (0 or 1 or 3)	-	-

Note: Before setting Ext_C(LCD_DISPCTL[16]), user needs to set LCD C-Type.

6.18.5 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
LCD Base Address: **LCD_BA = 0x400B_0000**				
LCD_CTL	LCD_BA+0x00	R/W	LCD Control Register	0x0000_0000
LCD_DISPCTL	LCD_BA+0x04	R/W	LCD Display Control Register	0x0000_0000
LCD_MEM_0	LCD_BA+0x08	R/W	LCD SEG3 ~ SEG0 data	0x0000_0000
LCD_MEM_1	LCD_BA+0x0C	R/W	LCD SEG7 ~ SEG4 data	0x0000_0000
LCD_MEM_2	LCD_BA+0x10	R/W	LCD SEG11 ~ SEG8 data	0x0000_0000
LCD_MEM_3	LCD_BA+0x14	R/W	LCD SEG15 ~ SEG12 data	0x0000_0000
LCD_MEM_4	LCD_BA+0x18	R/W	LCD SEG19 ~ SEG16 data	0x0000_0000
LCD_MEM_5	LCD_BA+0x1C	R/W	LCD SEG23 ~ SEG20 data	0x0000_0000
LCD_MEM_6	LCD_BA+0x20	R/W	LCD SEG27 ~ SEG24 data	0x0000_0000
LCD_MEM_7	LCD_BA+0x24	R/W	LCD SEG31 ~ SEG28 data	0x0000_0000
LCD_MEM_8	LCD_BA+0x28	R/W	LCD SEG35 ~ SEG32 data	0x0000_0000
LCD_FCR	LCD_BA+0x30	R/W	LCD frame counter control register	0x0000_0000
LCD_FCSTS	LCD_BA+0x34	R/W	LCD frame counter status	0x0000_0000

6.18.6 Register Description

LCD Control Register (LCD_CTL)

Register	Offset	R/W	Description	Reset Value
LCD_CTL	LCD_BA+0x00	R/W	LCD Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						PDINT_EN	PDDISP_EN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BLINK	FREQ			MUX			EN

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9]	**PDINT_EN**	**Power Down Interrupt Enable Control** If the power down request is triggered from system management, LCD controller will execute the frame completely to avoid the DC component. When the frame is executed completely, the LCD power down interrupt signal is generated to inform system management that LCD controller is ready to enter power down state, if PDINT_EN is set to 1. Otherwise, if PDINT_EN is set to 0, the LCD power down interrupt signal is blocked and the interrupt is disabled to send to system management. 0 = Power Down Interrupt Disabled. 1 = Power Down Interrupt Enabled.
[8]	**PDDISP_EN**	**Power Down Display Enable Control** The LCD can be programmed to be displayed or not be displayed at power down state by PDDISP_EN setting. 0 = LCD display Disabled (LCD is put out) at power down state. 1 = LCD display Enabled (LCD keeps the display) at power down state.
[7]	**BLINK**	**LCD Blinking Enable Control** 0 = Blinking Disabled. 1 = Blinking Enabled.
[6:4]	**FREQ**	**LCD Frequency Selection** 000 = LCD_CLK Divided by 32. 001 = LCD_CLK Divided by 64. 010 = LCD_CLK Divided by 96. 011 = LCD_CLK Divided by 128. 100 = LCD_CLK Divided by 192. 101 = LCD_CLK Divided by 256. 110 = LCD_CLK Divided by 384. 111 = LCD_CLK Divided by 512.

<table>
<tr><td>[3:1]</td><td><b>MUX</b></td><td><b>Mux Select</b>
000 = Static.
001 = 1/2 duty.
010 = 1/3 duty.
011 = 1/4 duty.
100 = 1/5 duty.
101 = 1/6 duty.
110 = Reserved.
111 = Reserved.

<b>Note:</b> User does not need to set PD_H_MFP bit field, but only to set the MUX bit field to switch LCD_SEG0 and LCD_SEG1 to LCD_COM4 and LCD_COM5.
<table>
<tr><td colspan="7"><b>LCD_SEG0(LCD_COM4) and LCD_SEG1(LCD_COM5) Pins Definition for Setting MUX Bit Field</b></td></tr>
<tr><td rowspan="2"><b>MUX</b></td><td colspan="2"><b>LQFP100</b></td><td colspan="2"><b>LQFP64</b></td><td colspan="2"><b>LQFP48</b></td></tr>
<tr><td>SEG0 (COM4) for PD.6</td><td>SEG1 (COM5) for PD.5</td><td>SEG0 (COM4) for PD.6</td><td>SEG1 (COM5) for PD.5</td><td>SEG0 (COM4) for PC.15</td><td>SEG1 (COM5) for PC.14</td></tr>
<tr><td>000, 001
010, 011</td><td>LCD_SEG 0</td><td>LCD_SEG 1</td><td>LCD_SEG 0</td><td>LCD_SEG 1</td><td>LCD_SEG 0</td><td>LCD_SEG 1</td></tr>
<tr><td>100</td><td>LCD_COM 4</td><td>LCD_SEG 1</td><td>LCD_COM 4</td><td>LCD_SEG 1</td><td>LCD_COM 4</td><td>LCD_SEG 1</td></tr>
<tr><td>101</td><td>LCD_COM 4</td><td>LCD_COM 5</td><td>LCD_COM 4</td><td>LCD_COM 5</td><td>LCD_COM 4</td><td>LCD_COM 5</td></tr>
</table></td></tr>
<tr><td>[0]</td><td><b>EN</b></td><td><b>LCD Enable Control</b>
0 = LCD controller operation Disabled.
1 = LCD controller operation Enabled.</td></tr>
</table>

LCD Display Control Register (LCD_DISPCTL)

Register	Offset	R/W	Description	Reset Value
LCD_DISPCTL	LCD_BA+0x04	R/W	LCD Display Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved					Res_Sel		Ext_C
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		CPUMP_FREQ			CPUMP_VOL_SET		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	BV_SEL	Reserved	IBRL_EN	Reserved	BIAS_SEL		CPUMP_EN

Bits	Description	
[31:19]	**Reserved**	Reserved.
[18:17]	**Res_Sel**	**R-type Resistor Value Selection** The LCD operation current will be different when we select different R-type resistor value. 00 = 200K Ohm. 01 = 300K Ohm. 10 = Reserved. 11 = 400K Ohm.
[16]	**Ext_C**	**Ext_C Mode Selection** This mode is similar to C-type LCD mode, but the operation current is lower than C-type mode. The control register setting is same with C-type mode except this bit is set to “1”. 0 = Disable. 1 = Enable.
[15:14]	**Reserved**	Reserved.
[13:11]	**CPUMP_FREQ**	**Charge Pump Frequency Selection** 000 = LCD_CLK. 001 = LCD_CLK/2. 010 = LCD_CLK/4. 011 = LCD_CLK/8. 100 = LCD_CLK/16. 101 = LCD_CLK/32. 110 = LCD_CLK/64. 111 = LCD_CLK/128.
[10:8]	**CPUMP_VOL_SET**	**Charge Pump Voltage Selection** 000 = 2.7V. 001 = 2.8V. 010 = 2.9V.

		011 = 3.0V. 100 = 3.1V. 101 = 3.2V. 110 = 3.3V. 111 = 3.4V.
[7]	**Reserved**	Reserved.
[6]	**BV_SEL**	**Bias Voltage Type Selection** 0 = C-Type bias mode. Bias voltage source from internal bias generator. 1 = R-Type bias mode. Bias voltage source from external bias generator. **Note:** The external resistor ladder should be connected to the V1 pin, V2 pin, V3 pin and V_{SS}. The V_{LCD} pin should also be connected to V_{DD}.
[5]	**Reserved**	Reserved.
[4]	**IBRL_EN**	**Internal Bias Reference Ladder Enable Control** 0 = Bias reference ladder Disabled. 1 = Bias reference ladder Dnabled.
[3]	**Reserved**	Reserved.
[2:1]	**BIAS_SEL**	**Bias Selection** 00 = Static. 01 = 1/2 Bias. 10 = 1/3 Bias. 11 = Reserved.
[0]	**CPUMP_EN**	**Charge Pump Enable Control** 0 = Disabled. 1 = Enabled.

LCD MEM_x Register (LCD_MEMORY_x) (x= 0 ~ 8)

Register	Offset	R/W	Description	Reset Value
LCD_MEM_0	LCD_BA+0x08	R/W	LCD SEG3 ~ SEG0 data	0x0000_0000
LCD_MEM_1	LCD_BA+0x0C	R/W	LCD SEG7 ~ SEG4 data	0x0000_0000
LCD_MEM_2	LCD_BA+0x10	R/W	LCD SEG11 ~ SEG8 data	0x0000_0000
LCD_MEM_3	LCD_BA+0x14	R/W	LCD SEG15 ~ SEG12 data	0x0000_0000
LCD_MEM_4	LCD_BA+0x18	R/W	LCD SEG19 ~ SEG16 data	0x0000_0000
LCD_MEM_5	LCD_BA+0x1C	R/W	LCD SEG23 ~ SEG20 data	0x0000_0000
LCD_MEM_6	LCD_BA+0x20	R/W	LCD SEG27 ~ SEG24 data	0x0000_0000
LCD_MEM_7	LCD_BA+0x24	R/W	LCD SEG31 ~ SEG28 data	0x0000_0000
LCD_MEM_8	LCD_BA+0x28	R/W	LCD SEG35 ~ SEG32 data	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		**SEG_(3+4x)**					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**SEG_(2+4x)**					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**SEG_(1+4x)**					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**SEG_(0+4x)**					

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:24]	**SEG_(3+4x)**	**SEG_(3+4x) DATA for COM5~COM0 (x = 0 ~ 8).** **For the LCD Display Memory MAP, please refer to Figure 6-54.**
[23:22]	**Reserved**	Reserved.
[21:16]	**SEG_(2+4x)**	**SEG_(2+4x) DATA for COM5~COM0(x= 0 ~ 8).** **For the LCD Display Memory MAP, please refer to Figure 6-54.**
[17:15]	**Reserved**	Reserved.
[14:8]	**SEG_(1+4x)**	**SEG_(1+4x) DATA for COM5~COM0 (x= 0 ~ 8).** **For the LCD Display Memory MAP, please refer to Figure 6-54.**
[7:6]	**Reserved**	Reserved.
[5:0]	**SEG_(0+4x)**	**SEG_(0+4x) DATA for COM5~COM0 (x= 0 ~ 8).** **For the LCD Display Memory MAP, please refer to Figure 6-54.**

LCD Frame Counter Register (LCD_FCR)

Register	Offset	R/W	Description	Reset Value
LCD_FCR	LCD_BA+0x30	R/W	LCD frame counter control register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**FCV**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FCV				**PRESCL**		**FCINTEN**	**FCEN**

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9:4]	**FCV**	**Frame Counter Top Value** These 6 bits contain the top value of the Frame counter.
[3:2]	**PRESCL**	**Frame Counter Pre-scaler Value** 00 = CLKframe/1. 01 = CLKframe/2. 10 = CLKframe/4. 11 = CLKframe/8.
[1]	**FCINTEN**	**LCD Frame Counter Interrupt Enable Control** 0 = Frame counter interrupt Disabled. 1 = Frame counter interrupt Enabled.
[0]	**FCEN**	**LCD Frame Counter Enable Control** 0 = Disabled. 1 = Enabled.

LCD Frame Counter Status Register (LCD_INTSTS)

Register	Offset	R/W	Description	Reset Value
LCD_FCSTS	LCD_BA+0x34	R/W	LCD frame counter status	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**PDSTS**	**FCSTS**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**PDSTS**	**Power-down Interrupt Status** 0 = Inform system manager that LCD controller is not ready to enter power-down state until this bit becomes 1 if power down is set and one frame is not executed completely. 1 = Inform system manager that LCD controller is ready to enter power-down state if power down is set and one frame is executed completely.
[0]	**FCSTS**	**LCD Frame Counter Status** 0 = Frame counter value does not reach FCV (Frame Count TOP value). 1 = Frame counter value reaches FCV (Frame Count TOP value). If the FCINTEN is s enabled, the frame counter overflow Interrupt is generated.

6.18.7 Application Circuit

External Resister ladder

1. Most commonly used for high V_{DD} voltages.

2. Uses inexpensive resistors to create the multilevel LCD voltages. Regardless of the number of pixels that are energized the current remains constant. The voltage at point V_{LCD} is typically tied to V_{DD}, either internally or externally

3 The resister values are determined by two factors.

a. Display quality

b. Power consumption

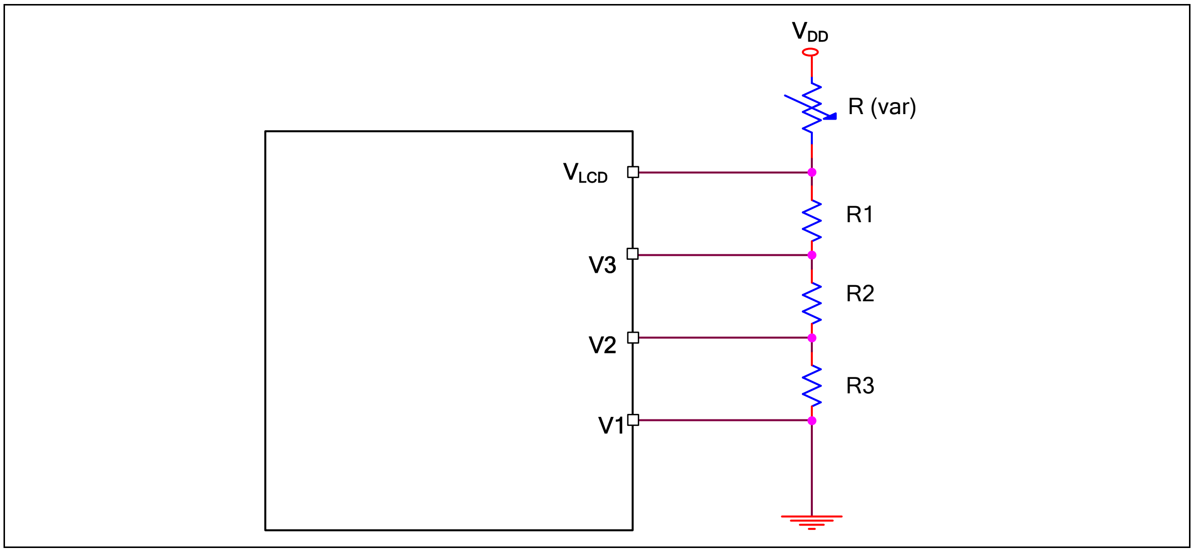


Figure 6-109 1/3 Bias (External Resistor Ladder)

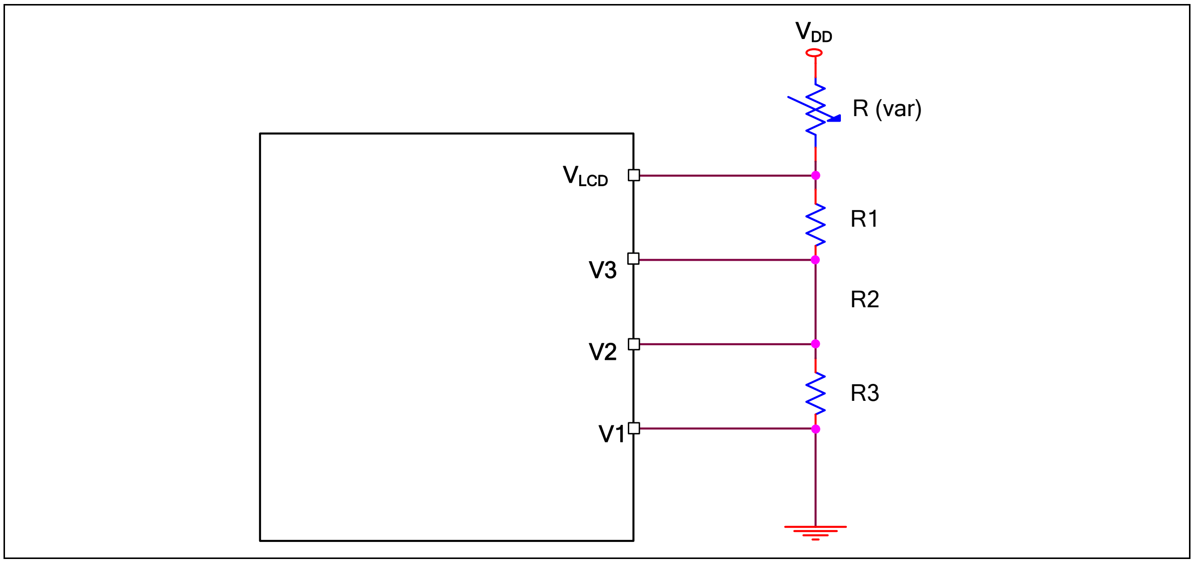


Figure 6-110 1/2 Bias (External Resistor Ladder)

Resistor ladder with capacitors

Sometimes the addition of parallel capacitors to the resistance can reduce the distortion caused by charging/discharging currents. This effect is limited since at some point a large resistor and large capacitor cause a voltage level shift which negatively impacts the display quality.

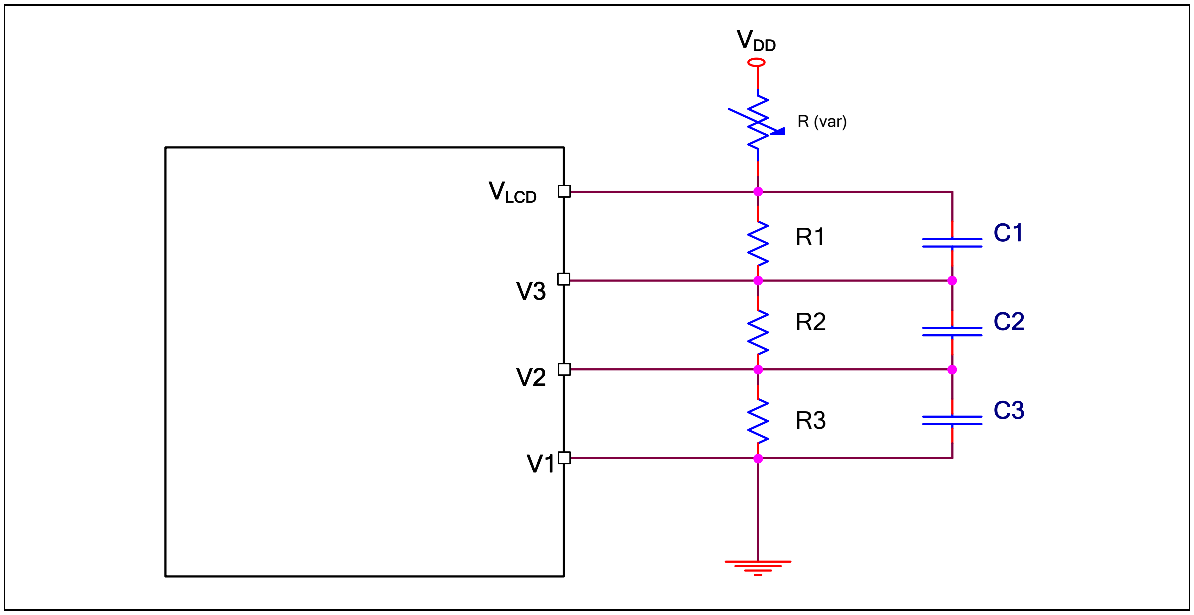


Figure 6-111 1/3 Bias (Resistor Ladder with Capacitor)

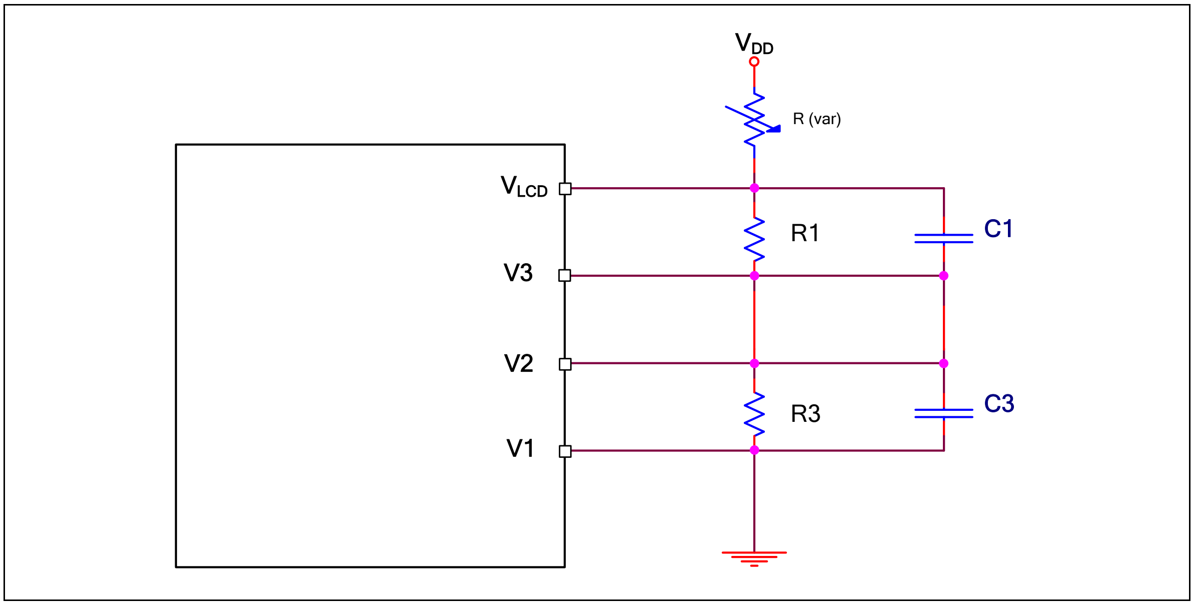


Figure 6-112 1/2 Bias (Resistor Ladder with Capacitor)

Internal R-type

Nano112 series MCUs also support external R-type mode (bypass internal R) to reduce current consumption. For external R-type application, VLCD is normally connected to system VDD, or it can be connected to VDD through an external variable resistor (VR) which is used for adjusting LCD contrast.

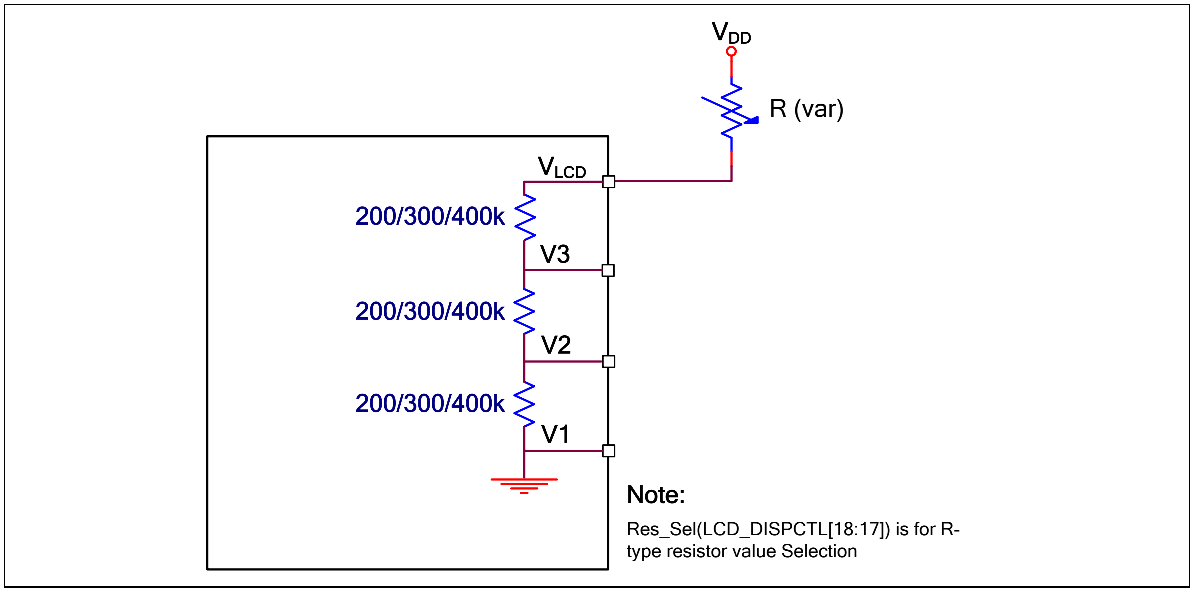

V_DD
R (var)
V_LCD
200/300/400k
V3
200/300/400k
V2
200/300/400k
V1
Note:
Res_Sel(LCD_DISPCTL[18:17]) is for R-type resistor value Selection

Figure 6‑113 1/2 Bias (Resistor Ladder with Capacitor)

Charge Pump

1. Ideal for low voltage battery operation because the V_{DD} voltage can be boosted up to drive the LCD panel.

2. The charge pump requires a charging capacitor and filter capacitor for each of the LCD voltages.

3. These capacitors are typically polyester, polypropylene, or polystyrene material.

4. Another feature that makes the charge pump ideal for battery applications is that the current consumption is proportional to the number of pixels that are energized.

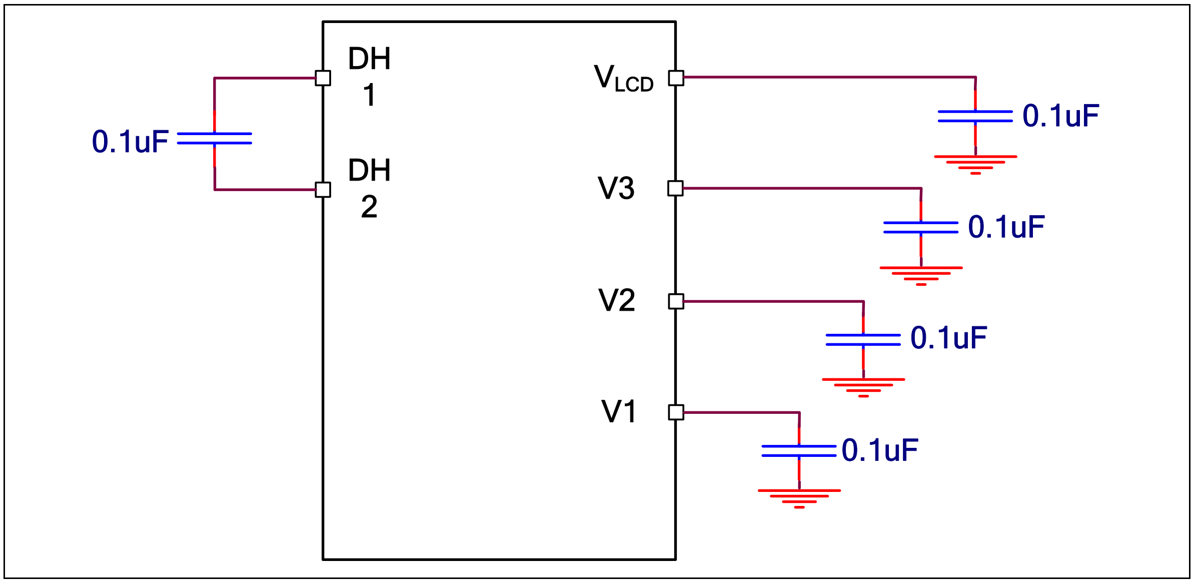


Figure 6‑114 1/3 Bias (Charge Pump)

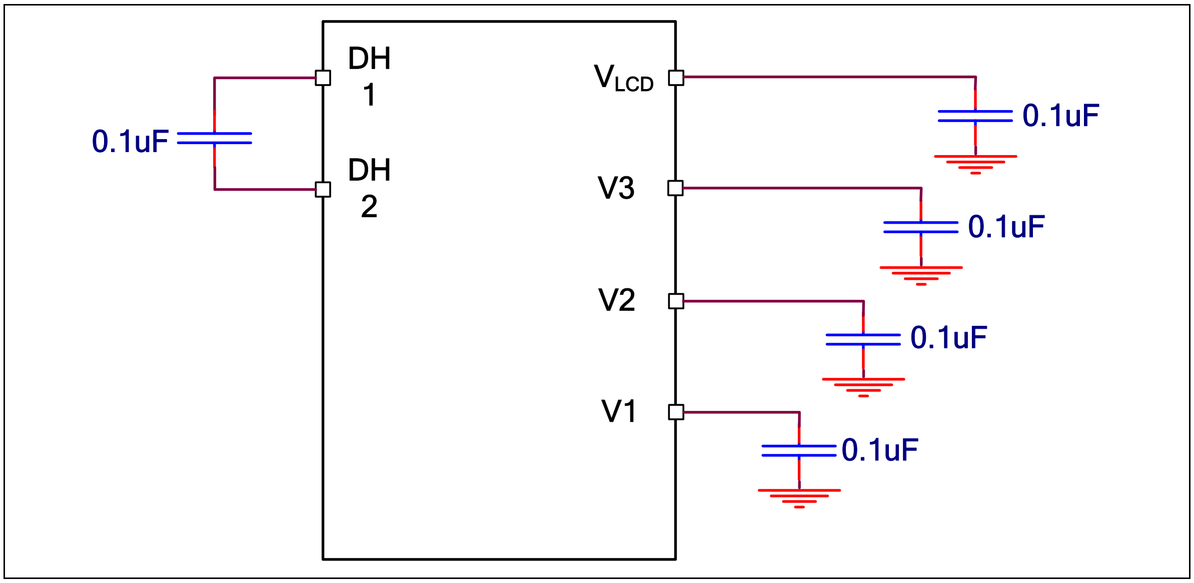
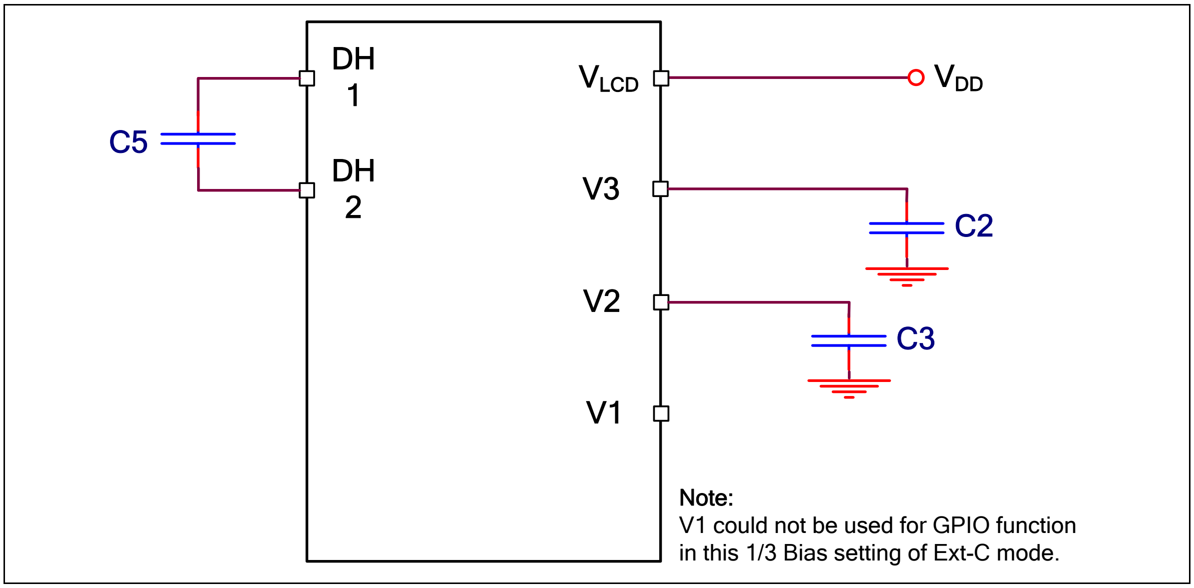


Figure 6-115 1/2 Bias (Charge Pump)

External Capacitor ladder

1. This mode is similar to C-type mode except the operation current is lower than C-type mode.

2. The major application circuit difference between C-type & Ext_C mode is that we can remove V1 capacitor in 1/3 bias application.



Note:
V1 could not be used for GPIO function in this 1/3 Bias setting of Ext-C mode.

Figure 6-64 1/3 Bias (External Capacitor Ladder)

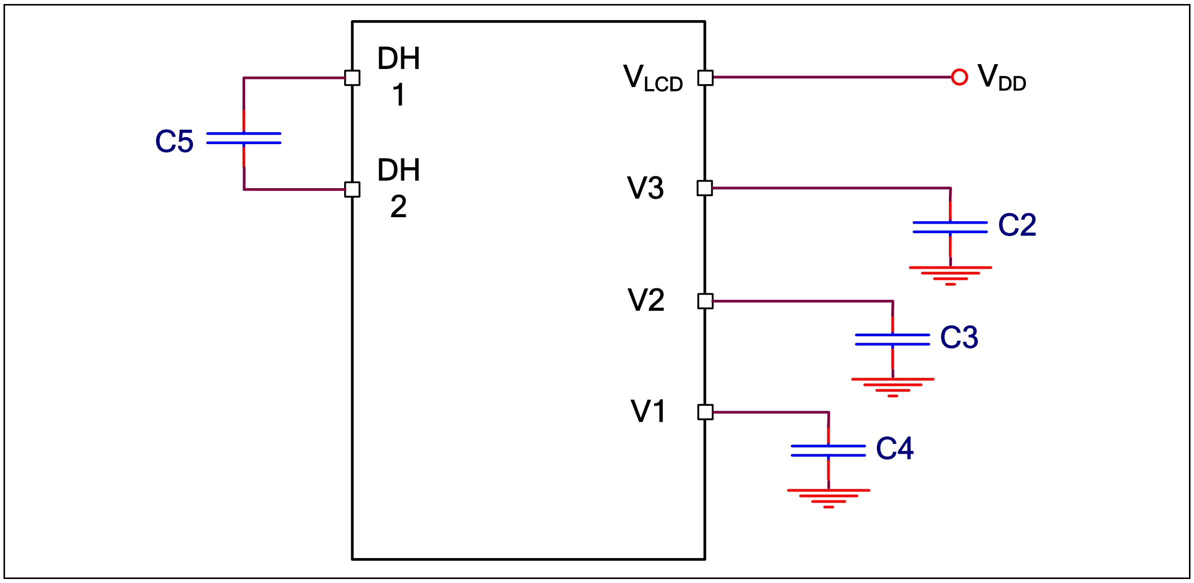


Figure 6‑65 1/2 Bias (External Capacitor Ladder)

6.19 Analog to Digital Converter (ADC)

6.19.1 Overview

The Nano112 series contains one 12-bit successive approximation analog-to-digital converter (SAR A/D converter) with 8 external input channels and 4 internal channels. The A/D converter supports three operation modes: Single, Single-cycle Scan and Continuous Scan mode, and can be started by software, external STADC(PA.11) pin, timer event start and PWM trigger.

Note that the I/O pins used as ADC analog input pins must configure the Pin Function (PA_L_MFP) to ADC input and off digital function (GPIOA_OFFD) should be turned on before ADC function is enabled.

6.19.2 Features

- Analog input voltage range: 0~V_{REF} (Max to AV_{DD})
- Selectable 12-bits, 10-bits, 8-bits and 6-bits resolution
- Supports sampling time settings for channel 0~7 individually (ADCCHSAMP0 register) and channel 14~17 share the same one sampling time setting (ADCCHSAMP1 register)
- Supports two Power-down modes:
  - Power-down mode
  - Standby mode
- Up to 8 external analog input channels (channel0 ~ channel7), and 4 internal channels (channel14~channel17) converting four voltage sources (internal reference voltage, internal temperature sensor output, AV_{DD}, and AV_{SS}).
- Maximum ADC clock frequency is 32 MHz and each conversion is 19 clocks+ sampling time depending on the input resistance (Rin).
- Three operating modes:
  - Single mode: A/D conversion is performed one time on a specified channel.
  - Single-cycle Scan mode: A/D conversion is performed one cycle on all specified channels with the sequence from the lowest numbered channel to the highest numbered channel.
  - Continuous Scan mode: A/D converter continuously performs Single-cycle scan mode until software stops A/D conversion.
- An A/D conversion can be started by:
  - Software write 1 to ADST bit
  - External pin STADC
  - PWM trigger
  - Selects one from four timer events (TMR0, TMR1, TMR2 and TMR3) that enable ADC and transfer AD results by PDMA
- Conversion results held in data registers for each channel
- Supports digital comparator: Conversion result can be compared with a specified value and user can select whether to generate an interrupt when conversion result is equal to the compare register setting.
- Supports Calibration and load Calibration words capability.

6.19.3 Block Diagram

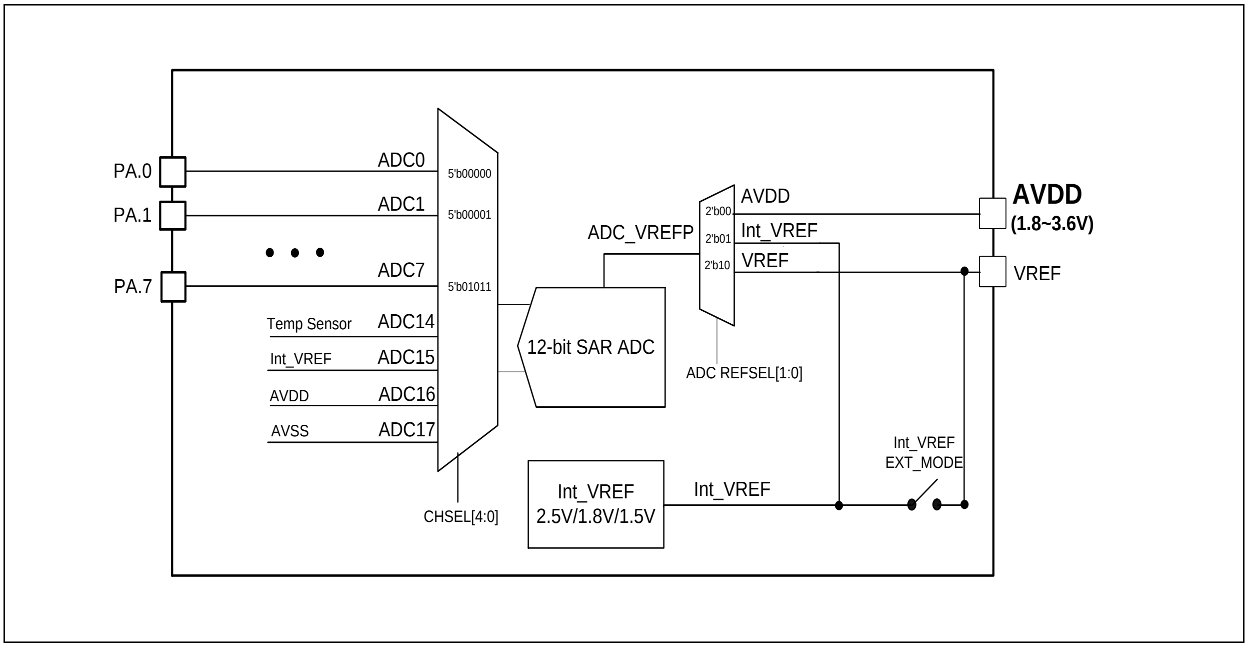


Figure 6‑116 ADC Block Diagram

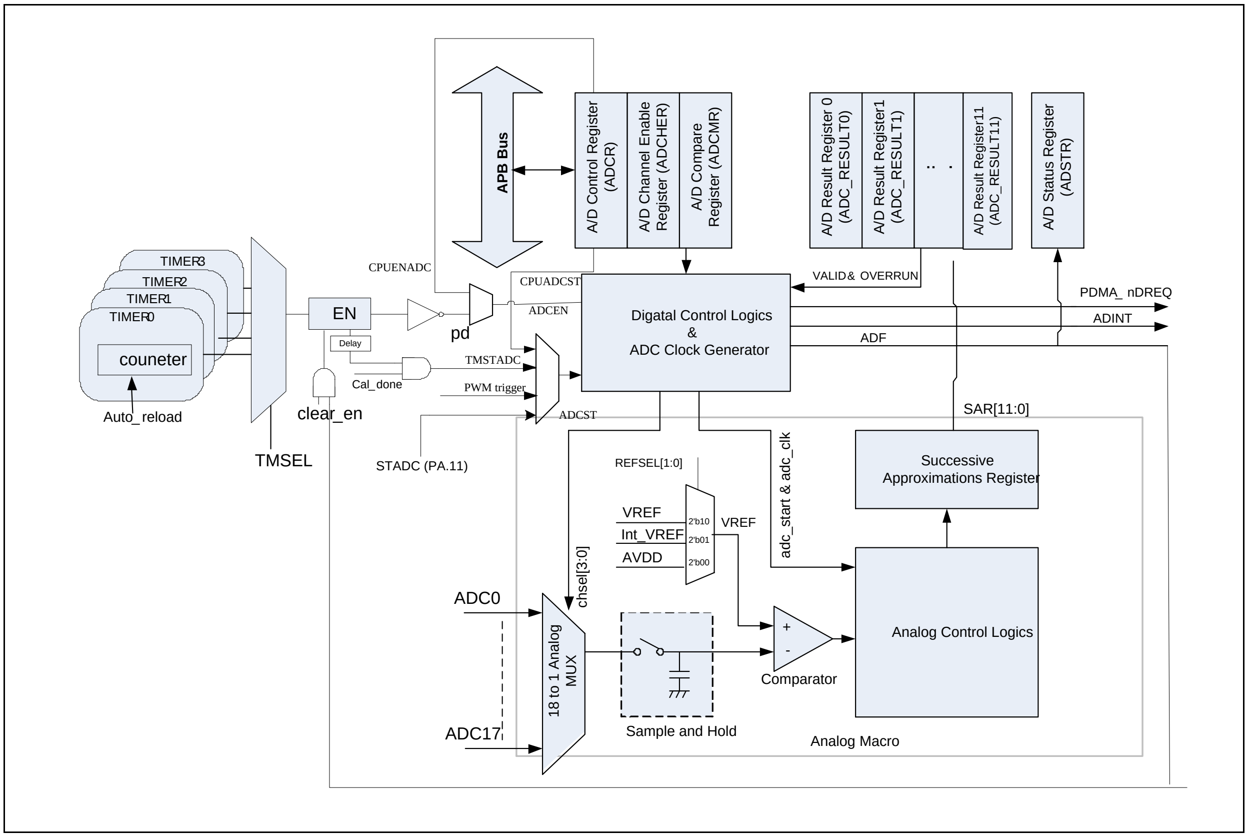


Figure 6‑117 ADC Controller Block Diagram

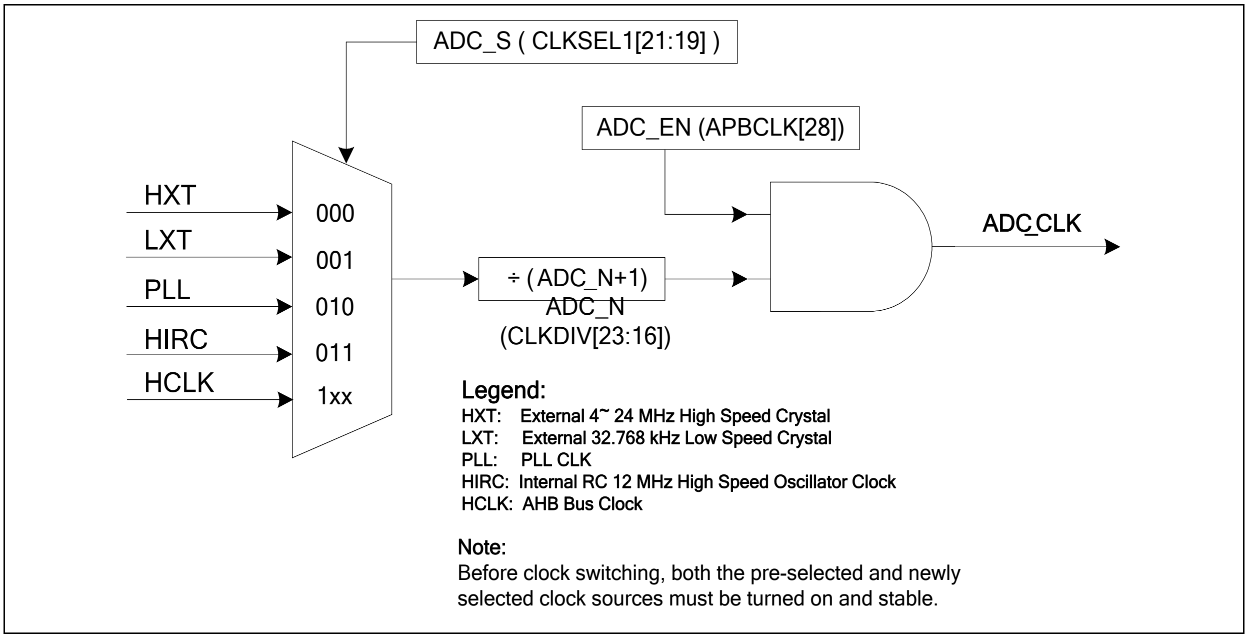

ADC_S (CLKSEL1[21:19])
ADC_EN (APBCLK[28])
HXT
LXT
PLL
HIRC
HCLK
000
001
010
011
1xx
÷ (ADC_N+1)
ADC_N
(CLKDIV[23:16])
ADC_CLK
Legend:
HXT: External 4~ 24 MHz High Speed Crystal
LXT: External 32.768 kHz Low Speed Crystal
PLL: PLL CLK
HIRC: Internal RC 12 MHz High Speed Oscillator Clock
HCLK: AHB Bus Clock
Note:
Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.

Figure 6-118 ADC Clock Control

6.19.4 Functional Description

The A/D converter is operated by successive approximation with 12-bit resolution. The ADC has three operation modes: Single mode, Single-cycle Scan mode and Continuous Scan mode. When changing the operating mode or analog input channel enabled, in order to avoid incorrect operation, software must clear the ADST (ADCR[11]) to 0. After the operation, the A/D converter discards current conversion and enters idle state while ADST (ADCR[11]) is cleared.

In some applications for saving power, ADC can be enabled by a time-out (TMRx Chy) signal and start A/D conversion after a delay time interval and enter power-down state after converting fixed amount of conversion data transferred to memory through PDMA, There are four time-out source(Timer0~3) to enable ADC by setting TMSEL (ADCR[13:12]) register.

6.19.4.1 Single Mode

In single mode, A/D conversion is performed only once on the specified single channel. The operations are as follows.

A/D conversion is started when the ADST (ADCR[11]) is set to 1 either by software or by external trigger input or by timer event selected by TMSEL (ADCR[13:12]) register or by PWM trigger.

When A/D conversion is finished, the result is stored in the A/D data register corresponding to the channel.

On completion of conversion, the ADF (ADSR[0]) is set to 1 and ADC interrupt is requested if the ADIE (ADCR[1]) is set to 1.

The ADST (ADCR[11]) remains 1 during A/D conversion. When A/D conversion ends, the ADST bit is automatically cleared to 0 and the A/D converter enters in idle state. If the ADST (ADCR[11]) is cleared to 0 by software during A/D conversion, A/D conversion will stop and enter in idle state.

Note: If user enables more than one channel in single mode, the least channel is converted and other enabled channels will be ignored.

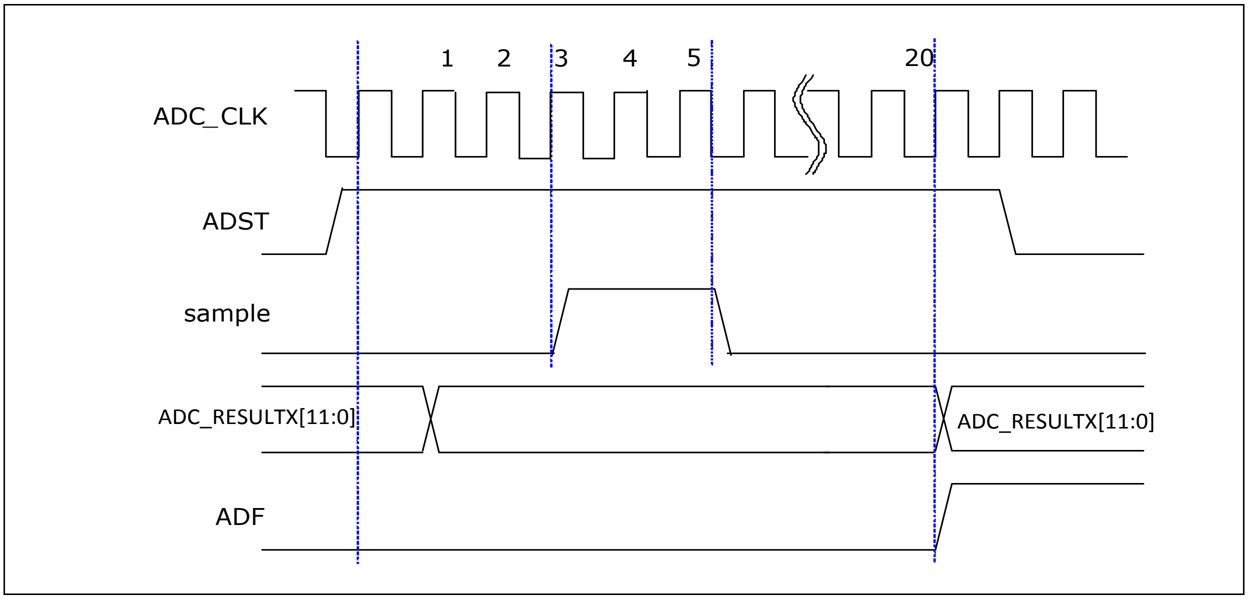


Figure 6-119 ADC Single Mode Conversion Timing Diagram

6.19.4.2 Single-Cycle Scan Mode

In single-cycle scan mode, A/D conversion will sample and convert the specified channels once in the sequence from the least numbered channel to the highest numbered channel. Operations are described as follows.

When the ADST (ADCR[11]) is set to 1 by software or by an external trigger input or by timer event selected by TMSEL (ADCR[13:12]) register or by PWM trigger, A/D conversion starts on the lowest numbered channel.

When A/D conversion for each enabled channel is completed, the result is sequentially transferred to the A/D data register corresponding to each channel.

When conversions of all the enabled channels are completed, the ADF (ADSR[0]) is set to 1. If the ADIE (ADCR[1]) is set to 1 at this time, an ADINT interrupt is set after A/D conversion ends.

After A/D conversion ends, the ADST (ADCR[11]) is automatically cleared to 0 and the A/D converter enters in idle state. If the ADST (ADCR[11]) is cleared to 0 by software during A/D conversion, A/D conversion will stop after current conversion complete and enter in idle state.

An example timing diagram for single-cycle scan is shown below:

(In this example, channel 0,2,3 and 7 are enabled.)

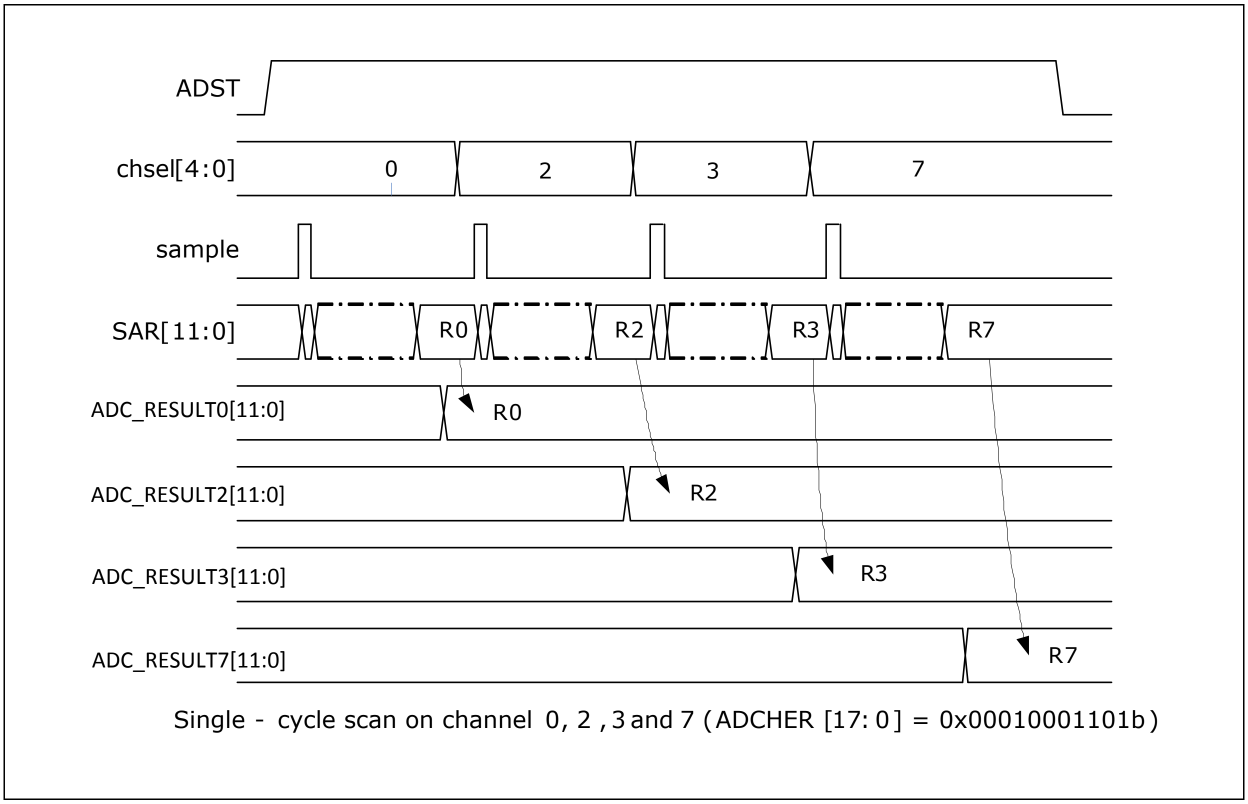


Figure 6-120 ADC Single-cycle Scan on Enabled Channels Timing Diagram

6.19.4.3 Continuous Scan Mode

In continuous scan mode, A/D conversion is performed sequentially on the specified channels that enabled by CHENx (ADCHER[17:0]) register (maximum 8 external channels and four internal channel for ADC). The operations are as follows.

1. When the ADST (ADCR[11]) is set to 1 by software or external trigger input or by timer event selected by TMSEL (ADCR[13:12]) register or by PWM trigger, A/D conversion starts on the channel with the lowest number.
2. When A/D conversion for each enabled channel is completed, the result of each enabled channel is stored in the A/D data register corresponding to each enabled channel.
3. Once when all of the enabled channel sequentially completes A/D converting, the ADF (ADSR[0]) will be set to 1. If the ADIE (ADCR[1]) is set to 1 at this time, an ADINT interrupt is set after A/D conversion ends.
4. Follow Step 3, conversion of the first enabled channel starts again.

Steps 2 to 4 are repeated as long as the ADST (ADCR[11]) remains set to 1. When the ADST (ADCR[11]) is cleared to 0, A/D conversion stops and the A/D converter enters the idle state.

An example timing diagram for continuous scan on enabled channels (0, 2, 3 and 7) is shown below:

(In this example, channel 0, 2, 3 and 7 are enabled.)

(This example is only appropriate for ADC.)

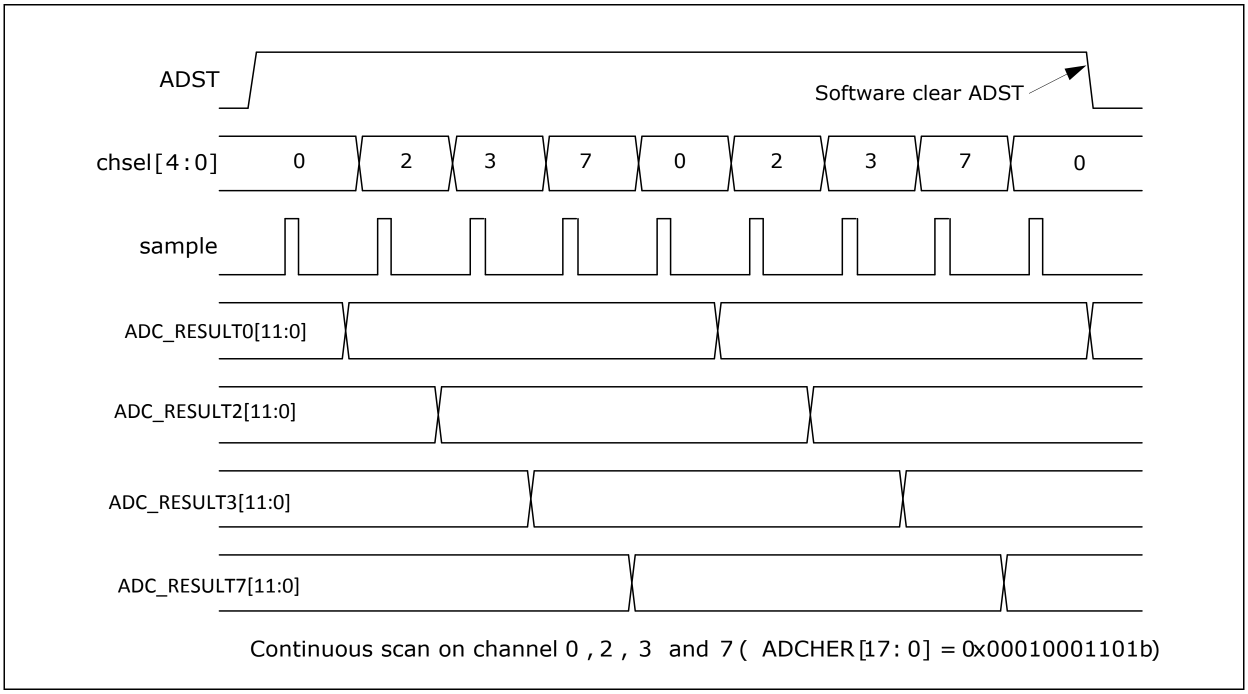


Figure 6-121 ADC Continuous Scan on Enabled Channels Timing Diagram

6.19.4.4 ADC Started by External Triggering

A/D conversion can be triggered by external pin request. When the TRGE (ADCR[8]) is set to high to enable ADC external trigger function, setting the TRGS (ADCR[5:4]) to 00b is to select external trigger input from the STADC pin. Software can set TRGCOND (ADCR[7:6]) to select trigger condition is falling/rising edge or low/high level. If level trigger condition is selected, the STADC pin must be kept at defined state at least 8 PCLKs. The ADST (ADCR[11]) will be set to 1 at the 9th PCLK and start to conversion. In level trigger mode conversion is continuous as long as the external trigger input is in asserted state if external trigger input is pull at low (or high state). It is stopped only when external condition trigger condition disappears. If edge trigger condition is selected, the high and low state must be kept at least 4 PLCKs. When a trigger signal with pulse width smaller than the specified width (4 PCLKs), conversion is not triggered.

6.19.4.5 Conversion Result Monitor by Compare Mode

The ADC controller provides two sets of compare registers ADCMPR0 and ADCMPR1 to monitor at most two specified channel conversion results from A/D conversion module (refer to Figure 6-122). Software can select which channel to be monitored by setting CMPCH(ADCMPRx[5:0]) and CMPCOND (ADCMPRx[2]) which is used to check conversion result is either less or greater than (equal to) the specified value in CMPD(ADCMPRx[27:16]). When the conversion of the channel specified by CMPCH (ADCMPRx[7:3]) is completed, the comparing action will be triggered one time automatically. If the compare result meets the setting, compare match counter will increase by 1, Once the counter value reaches the setting of (CMPMATCNT (ADCMPRx[11:8])+1), CMPFx (ADSR[2,1]) will be set to 1. If the compare result does not meet the setting, compare match counter will reset to 0. If CMPIE (ADCMPRx[1]) is set, an ADINT interrupt request is generated. Software can use this function to monitor whether an external analog input voltage traverse the specified threshold in scan mode without imposing a load on software. Detailed logics diagram is shown below.

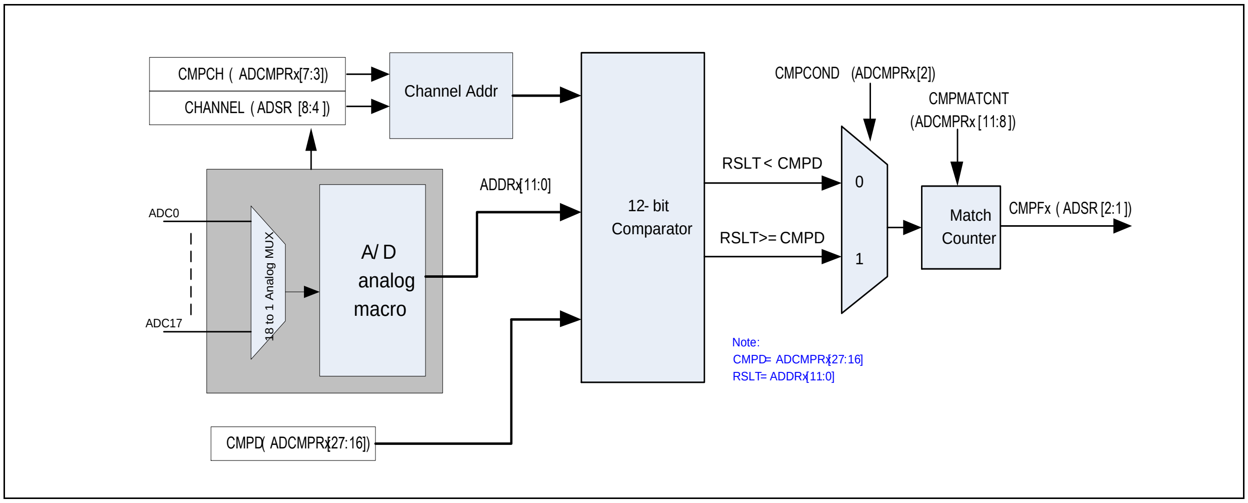


Figure 6‑122 ADC Conversion Result Monitor Logic Diagram

6.19.4.6 Interrupt Sources

The A/D converter generates a conversion end flag, ADF (ADSR[0]) register at the ending moment of A/D conversion. If ADIE (ADCR[1]) is set, the conversion end interrupt is asserted via ADINT occurs. If CMPIE bit is enabled and A/D conversion result meets the setting in ADCMPR register, monitor interrupt occurs, and ADINT will be set also. CPU can clear CMPF and ADF to stop interrupt request.

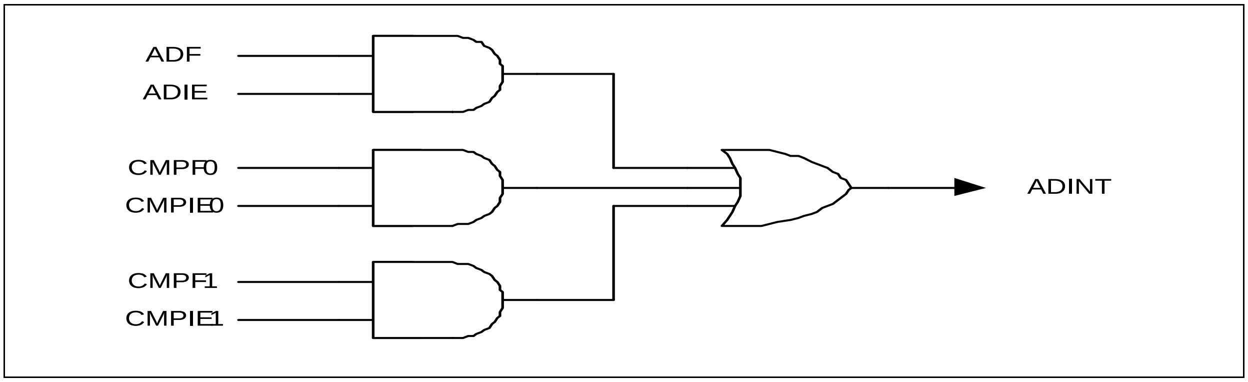


Figure 6‑123 ADC Controller Interrupt

6.19.4.7 Peripheral DMA Request

When A/D conversion is finished, the converted result is loaded into ADC_RESULTx(x=0~7) register and VALID bit is set to 1. If PTEN (ACDR[9]) is set, ADC controller will generate PDMA request to ask a data transfer. Having the converted result read by PDMA in response to PDMA request enables continuous conversion to be achieved without CPU intervention.

6.19.4.8 ADC enabled by timer event

User can configure ADC to use timer trigger function by programming TMSEL (ACDR[13:12]), TMTRGMOD (ACDR[15]) and TMPDMACNT (ACDR[31:24]). If AD is power down, timer event can enable ADC. TMSEL (ACDR[13:12]) selects timer event source.

After ADC is woken up, it starts to transfer and pass the ADC_RESUT to memory by PDMA. User should configure PTEN (ACDR[9]) register to enable PDMA transfer and configure ADMD (ACDR[3:2]) to run ADC in continuous, single or single cycle mode; and configure TMPDMACNT (ACDR[31:24]) register to specify the amount of ADC_RESULT that PDMA will deliver to memory each time the timer event occurs. After PDMA have delivered the amount of ADC_RESULT specified in TMPDMACNT (ACDR[31:24]), ADC will go to power down until the next timer event coming.

After the total amount of ADC_RESULT configured in PDMA byte count register have been delivered to memory, ADC will go to power down, this time the ADC will not be waken up by the following timer event any more.

All the configurations should be done before the system entering power down because CPU can't read and write register while the whole system is power down. In single-cycle and single mode, PDMA transfer count should be exact the same with enabled channel count.

6.19.4.9 ADC enabled by PWM trigger

A/D conversion can be triggered by PWM. When the TRGE (ADCR[8]) is set to high to enable ADC external trigger function, setting the TRGS (ADCR[5:4]) to 11b is to select PWM trigger.

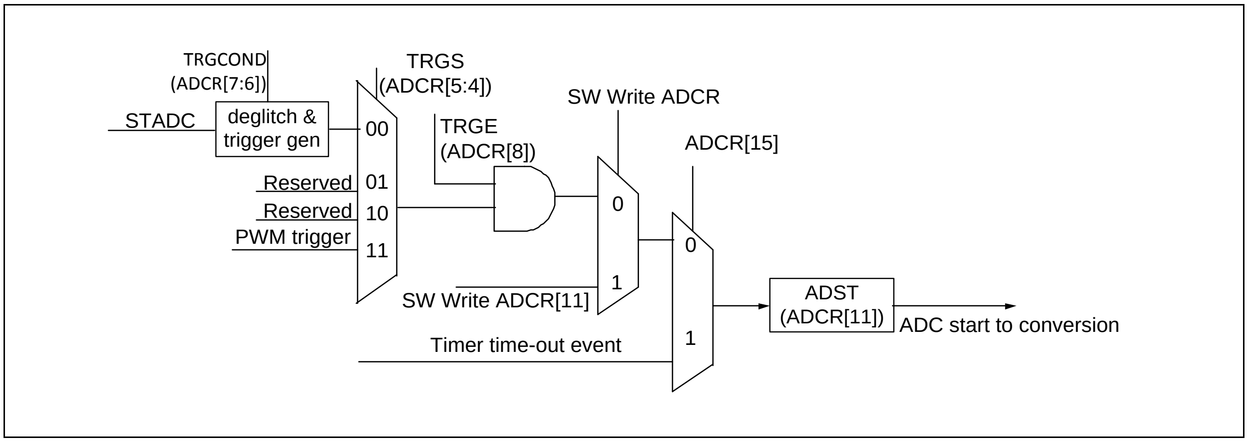


Figure 6-124 ADC Start Conversion Conditions

6.19.4.10 ADC sampling time

The figure below shows the (simplified) equivalent circuit of the S/H (sample and hold) input network, where C_S is the storage capacitor, R_S is the resistance of the sampling switch and R_I is the output impedance of the signal source V_I. The Figure 6-125 shows the situation where the conversion cycle j+1 starts immediately after conversion cycle n ends. In this case the duration of the sampling phase is, approximately, 1.5 x ADC_CLK. C_S must be charged in that phase, and it must be ensured that the voltage at its terminals becomes sufficiently near V_I. To guarantee this, R_I may not take arbitrarily large values

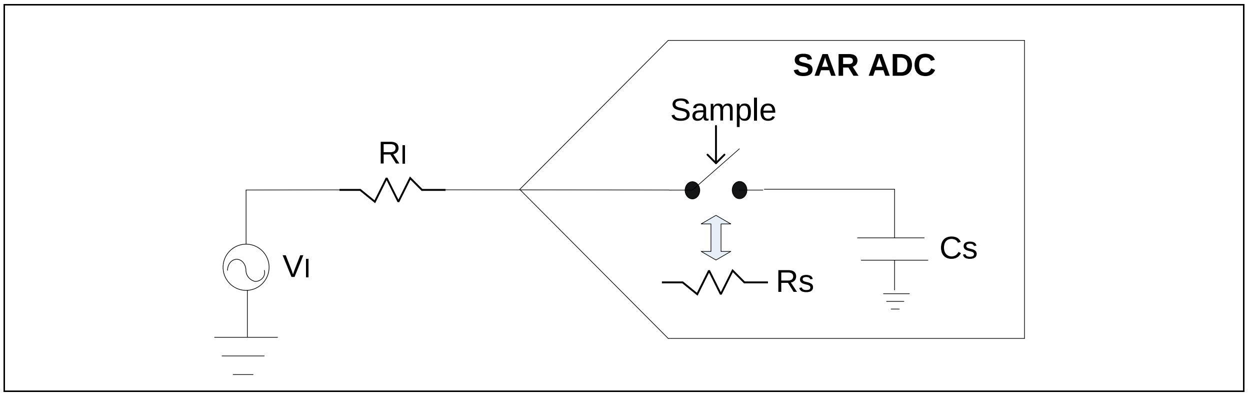


Figure 6-125 Model of Sampling Network

Figure 6‑126 shows how the sampling time can be increased, to allow the operation with signal sources having a low driving capability: the adc_start signal is delayed during the number of clock cycles necessary to guarantee the accurate input signal sampling. During this period the chsel must remain unchanged. Note that this operation reduces the effective sampling rate.

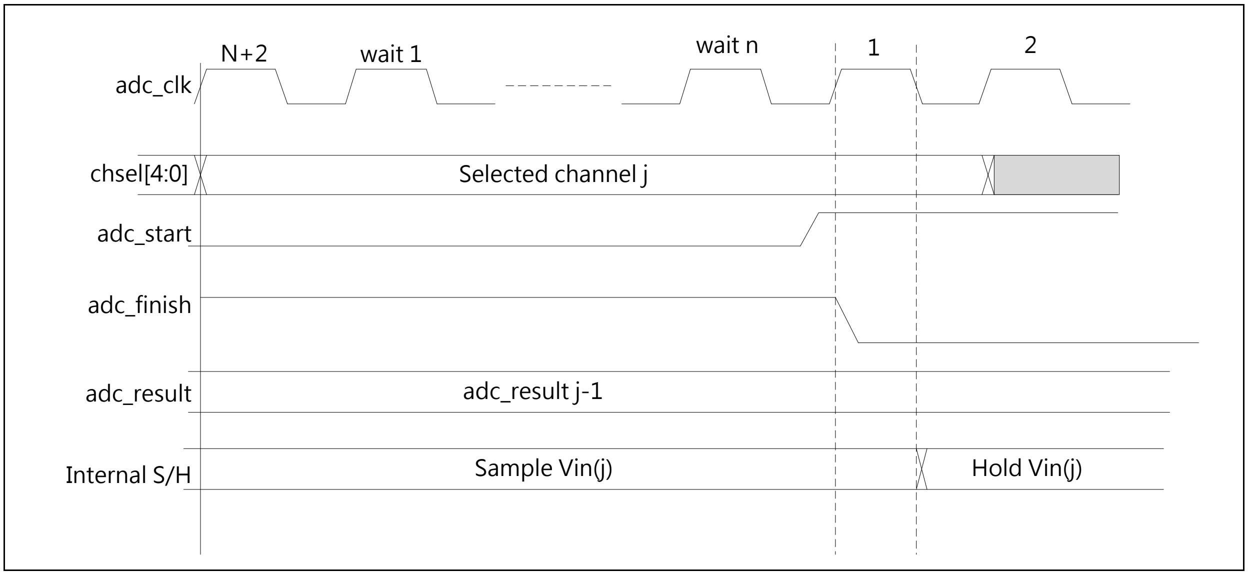


Figure 6‑126 Increased Sampling Time Waveform

For both types of inputs, it is possible to achieve the maximum sampling frequency, but under certain conditions (depending either from the resolution mode (*RESSEL*) or from the output impedance of the signal source) the sampling period should be delayed during the necessary clock cycles to guarantee the sampling precision (above figure).

The following graphics indicates the number of additional sample and hold cycles (n), necessary for a wide range of Ri (signal source output resistance) values, for all ADC input channels depending on the resolution mode (*RESSEL*). Use sampling counter registers (ADCCHSAMP0 and ADCCHSAMP1) to add additional sample and hold cycle (n).

Please note that these graphics refer to the **additional** sampling and hold clock cycles (i.e. in the situations where n=0, the sampling period is 1.5 × *ADC_CLK*).

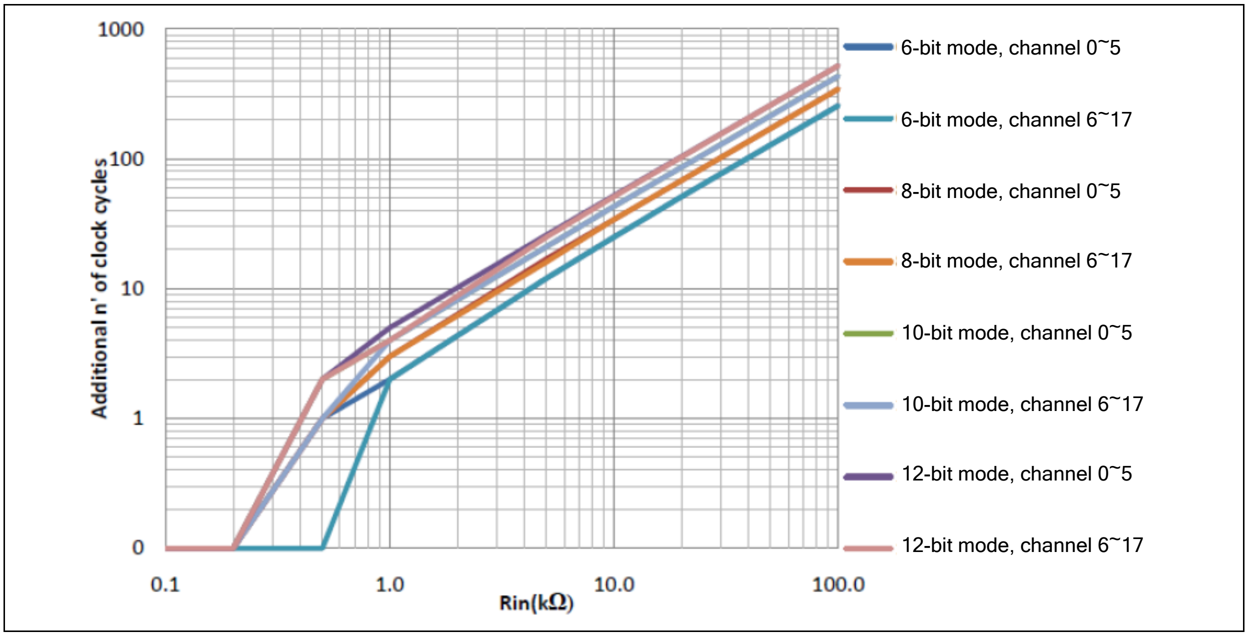


Figure 6-127 Additional Sample and Hold Clock Cycles (n) as a Function of the Signal Source Output Resistance Rin (kΩ)

The results presented in the graphic above were measured under normal conditions (typical process corner, V_{DD} = AV_{DD} = 3.3V, LDO output = 1.8V, Tjunction= 50°C, ADC_CLK=32 MHz, V_{REF}= AV_{DD}).

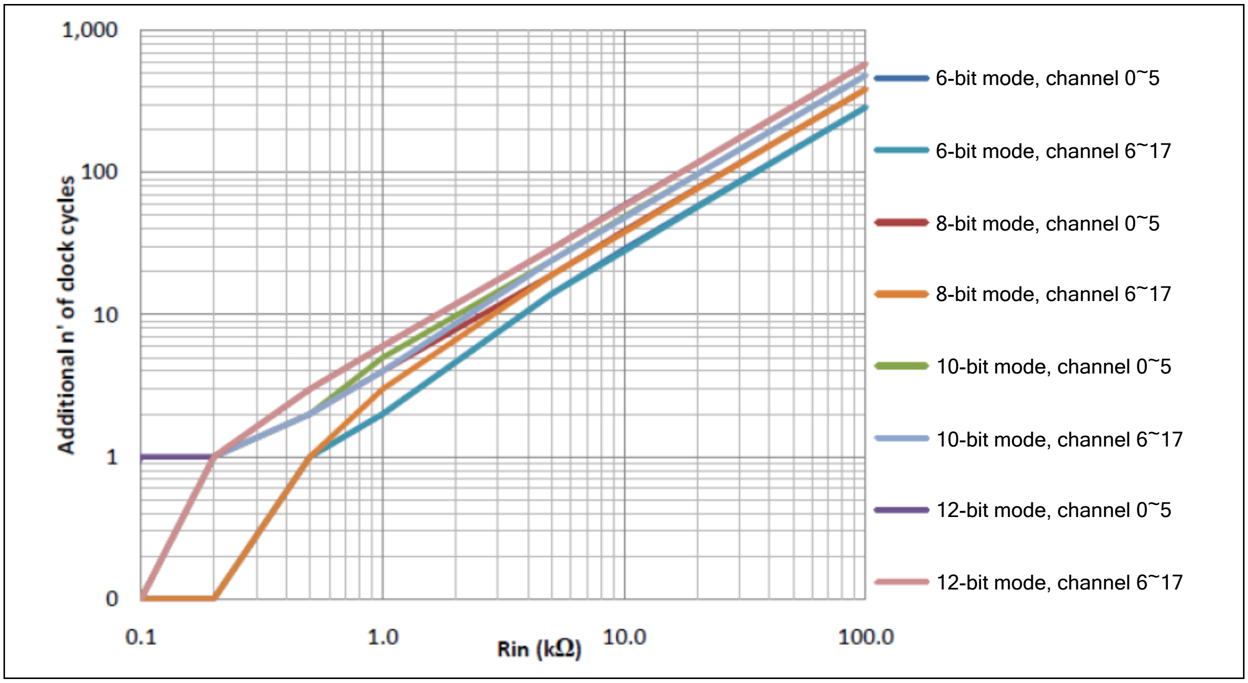


Figure 6-128 Additional Sample and Hold Clock Cycles (n) as a Function of the Signal Source Output Resistance Rin (kΩ)

The results presented in the graphic above were measured under the worst case conditions (slow process corner, V_{DD} = AV_{DD} = 1.8V, LDO output = 1.62V, Tjunction= -40°C, ADC_CLK=32 MHz, V_{REF}= AV_{DD}).

6.19.4.11 ADC Power-down mode

There are two Power-down modes user can select, including Power-down mode, and Standby mode. User can configure PWDMOD (ADCPWD[3:2]) to determine what Power-down mode that user want to be before disabling ADEN (ADCR[0]) register.

In different Power-down mode (power down, standby), the power up sequence are quite different, user should know currently Power-down mode and configure PWDMOD (ADCPWD[3:2]) to determine what power up sequence that user want to be before enabling ADEN (ADCR[0]), if the sequence was wrong, ADC would be mal-function .

The difference between those Power-down modes are power consumption and the stable time after resuming from each Power-down mode, the least power consumption is Power-down mode and then standby mode, and stable time are in reversed order Before ADC entering power down, make sure that ADC is stop (by disabling ADST) and all conversion are completed (by polling ADF (ADSR[0])).

6.19.4.12 ADC Offset calibration

To decrease the effect of electrical random noise, the ADC performs calibration to get average offset measurement. Afterwards, in normal operation, the digital block applies the calibrated word to the internal ADC capacitor array, so that the offset voltage is removed.

User can set CALEN (ADCCALCTL[0]) to high and select CALSEL (ADCCALCTL[3]) to 1(to do calibration) and write 1 to CALSTART (ADCCALCTL[1]), and then waiting for CALDONE (ADCCALCTL[2]) to high; when CALDONE (ADCCALCTL[2]) is high, the calibration is complete and the calibration word is in ADC_CALWORD register.

User can also load the specified calibrated word to ADC_CALWORD register to save time to complete the calibration method. The configuration are the same except setting CALSEL to 0, and waiting for CALDONE bit to high; when CALDONE is high, the load calibration word is complete and the loaded calibration word is now applied to ADC.

6.19.4.13 Selectable resolution

The ADC has the selectable resolution between 12, 10, 8 and 6 bit. User can choose the resolution by setting RESSEL (ADCR[17:16]).

The resolution selection can only be updated after the end of the conversion (ADF (ADSR[0]) becomes high), different resolutions will result in the different conversion cycle. Take 12 bit resolution for example, it will take 19 ADC clock cycle to complete one channel conversion; and 17 ADC clock cycle,15 ADC clock cycle, 14 ADC clock cycle for the resolution 10 bits, 8 bits, and 6 bits.

6.19.4.14 Temperature Sensor

The figure below shows the typical temperature sensor transfer function. The formula for the output voltage (V_{TEMP}) is as below equation.

V_{TEMP} (mV) = -1.73 (mV/°C) x Temperature (°C) + 740 (mV).

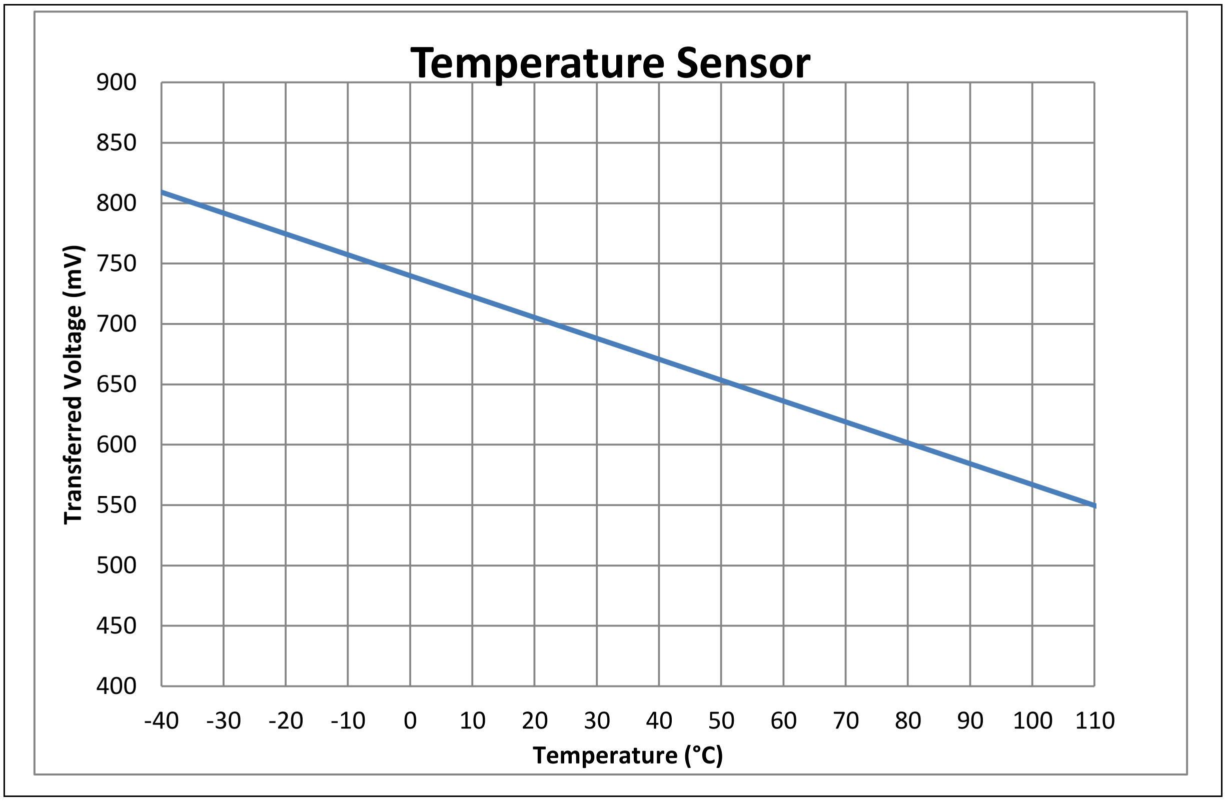


Figure 6‑129 Temperature Sensor Transfer Function

6.19.4.15 Internal Reference Voltage

The internal reference voltage (Int_V_{REF}) is an internal fixed reference voltage regardless of power supply variations. The Int_V_{REF} output is internally connected to ADC input channel15 (ADC15) and Analog Comparator's negative input side. For battery power detection application, user can connect V_{REF} pin to AV_{DD} for ADC reference voltage. The Int_V_{REF} can be used as ADC input channel such that user can convert the ADC value to estimate AV_{DD} voltage with following formula and the block diagram is shown as Figure 6‑130.

AV_{DD} = ((2^ N)/R)* Int_V_{REF}

N: ADC resolution

R: ADC conversion result

Int_V_{REF}: Internal reference voltage

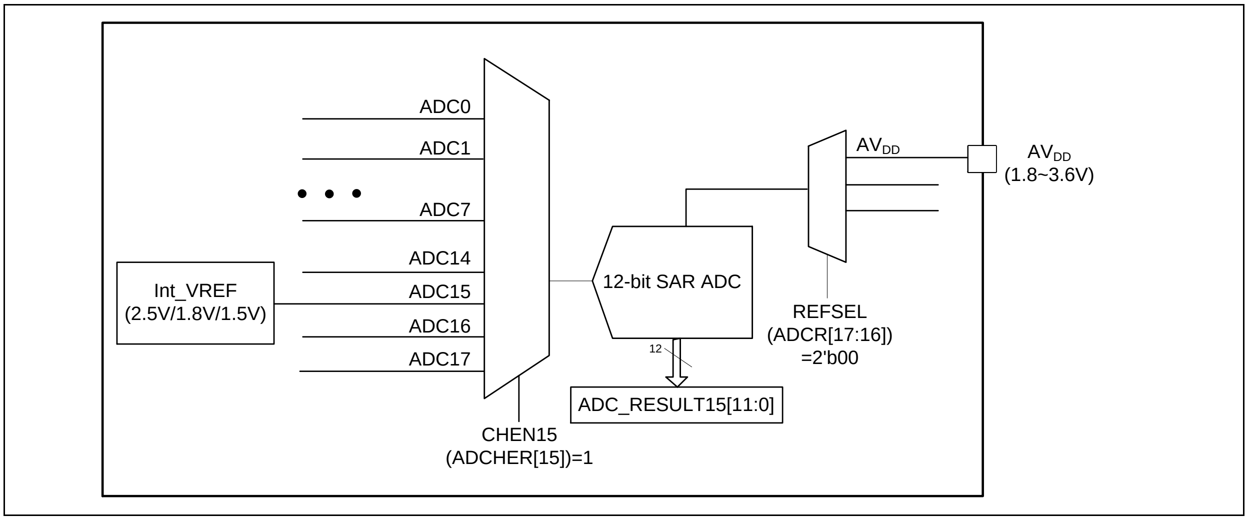


Figure 6-130 Int_V_{REF} for Measuring AV_{DD} Application Block Diagram

For example, The setting value for Int_V_{REF} is 1.8 V, the ADC is 12-bit resolution, and Int_V_{REF} is set as ADC input channel to trigger ADC conversion. Assuming ADC conversion result is 2048.

AV_{DD} = 3.6 V

N = 12

R = 2048

Int_V_{REF} = 1.8 V

AV_{DD} = ((2^ 12)/2048)* 1.8 = (4096/2048)* 1.8 = 3.6 V

If the ADC conversion result is 2457

AV_{DD} = ((2^ 12)/2458)* 1.8 = (4096/2458)* 1.8 = 3 V

If the ADC conversion result is 2949

AV_{DD} = ((2^ 12)/2949)* 1.8 = (4096/2949)* 1.8 = 2.5 V

6.19.5 Register Map

R: read only, **W**: write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
ADC Base Address: **ADC_BA = 0x400E_0000**				
ADC_RESULT0	ADC_BA+0x00	R	A/D Data Register 0	0x0000_0000
ADC_RESULT1	ADC_BA+0x04	R	A/D Data Register 1	0x0000_0000
ADC_RESULT2	ADC_BA+0x08	R	A/D Data Register 2	0x0000_0000
ADC_RESULT3	ADC_BA+0x0C	R	A/D Data Register 3	0x0000_0000
ADC_RESULT4	ADC_BA+0x10	R	A/D Data Register 4	0x0000_0000
ADC_RESULT5	ADC_BA+0x14	R	A/D Data Register 5	0x0000_0000
ADC_RESULT6	ADC_BA+0x18	R	A/D Data Register 6	0x0000_0000
ADC_RESULT7	ADC_BA+0x1C	R	A/D Data Register 7	0x0000_0000
ADC_RESULT14	ADC_BA+0x38	R	A/D Data Register 14	0x0000_0000
ADC_RESULT15	ADC_BA+0x3C	R	A/D Data Register 15	0x0000_0000
ADC_RESULT16	ADC_BA+0x40	R	A/D Data Register 16	0x0000_0000
ADC_RESULT17	ADC_BA+0x44	R	A/D Data Register 17	0x0000_0000
ADCR	ADC_BA+0x48	R/W	A/D Control Register	0x0001_0000
ADCHER	ADC_BA+0x4C	R/W	A/D Channel Enable Register	0x0000_0000
ADCMPR0	ADC_BA+0x50	R/W	A/D Compare Register 0	0x0000_0000
ADCMPR1	ADC_BA+0x54	R/W	A/D Compare Register 1	0x0000_0000
ADSR	ADC_BA+0x58	R/W	A/D Status Register	0x0000_0000
ADPDMA	ADC_BA+0x60	R	A/D PDMA current transfer data Register	0x0000_0000
ADCPWD	ADC_BA+0x64	R/W	ADC Power Management Register	0x0000_0A00
ADCCALCTL	ADC_BA+0x68	R/W	ADC Calibration Control Register	0x0000_0009
ADCCALWORD	ADC_BA+0x6C	R/W	A/D calibration load word register	0xXXXX_XXXX
ADCCHSAMP0	ADC_BA+0x70	R/W	ADC Channel Sampling Time Counter Register Group 0	0x0000_0000
ADCCHSAMP1	ADC_BA+0x74	R/W	ADC Channel Sampling Time Counter Register Group 1	0x0000_0000

6.19.6 Register Description

A/D Data Registers (ADC_RESULT0~ ADC_RESULT17)

Register	Offset	R/W	Description	Reset Value
ADC_RESULT0	ADC_BA+0x00	R	A/D Data Register 0	0x0000_0000
ADC_RESULT1	ADC_BA+0x04	R	A/D Data Register 1	0x0000_0000
ADC_RESULT2	ADC_BA+0x08	R	A/D Data Register 2	0x0000_0000
ADC_RESULT3	ADC_BA+0x0C	R	A/D Data Register 3	0x0000_0000
ADC_RESULT4	ADC_BA+0x10	R	A/D Data Register 4	0x0000_0000
ADC_RESULT5	ADC_BA+0x14	R	A/D Data Register 5	0x0000_0000
ADC_RESULT6	ADC_BA+0x18	R	A/D Data Register 6	0x0000_0000
ADC_RESULT7	ADC_BA+0x1C	R	A/D Data Register 7	0x0000_0000
ADC_RESULT14	ADC_BA+0x38	R	A/D Data Register 14	0x0000_0000
ADC_RESULT15	ADC_BA+0x3C	R	A/D Data Register 15	0x0000_0000
ADC_RESULT16	ADC_BA+0x40	R	A/D Data Register 16	0x0000_0000
ADC_RESULT17	ADC_BA+0x44	R	A/D Data Register 17	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						**OVERRUN**	**VALID**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**RSLT**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RSLT							

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17]	**OVERRUN**	**Over Run Flag** When VALID is high and ADC converts finish, this field will set to high.
[16]	**VALID**	**Data Valid Flag** After ADC converts finish, this field will set to high. This field will clear when this register be read.
[15:12]	**Reserved**	Reserved.

Bits	Description	
[11:0]	**RSLT**	**A/D Conversion Result** This field contains 12 bits conversion results.

A/D Control Register (ADCR)

Register	Offset	R/W	Description	Reset Value
ADCR	ADC_BA+0x48	R/W	A/D Control Register	0x0001_0000

31	30	29	28	27	26	25	24
TMPDMACNT							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				**RESSEL**		**REFSEL**	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TMTRGMOD	**Reserved**	**TMSEL**		**ADST**	**DIFF**	**PTEN**	**TRGEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TRGCOND		**TRGS**		**ADMD**		**ADIE**	**ADEN**

Bits	Description	
[31:24]	**TMPDMACNT**	**PDMA Count** When each timer event occur PDMA will transfer TMPDMACNT +1 ADC result in the amount of this register setting **Note:** The total amount of PDMA transferring data should be set in PDMA byte count register. When PDMA finish is set, ADC will not be enabled and start transfer even though the timer event occurred
[23:20]	**Reserved**	Reserved.
[19:18]	**RESSEL**	**Resolution Selection** 00 = 6 bits. ADC result will put at RSLT[5:0] (ADC_RESULTx[5:0]), 01 = 8 bits. ADC result will put at RSLT[7:0] (ADC_RESULTx[7:0]) 10 = 10 bits. ADC result will put at RSLT[9:0] (ADC_RESULTx[9:0]) 11 = 12 bits. ADC result will put at RSLT (ADC_RESULTx[11:0])
[17:16]	**REFSEL**	**Reference Voltage Source Selection** 00 = Select AV_{DD} as reference voltage. 01 = Select Int_V_{REF} as reference voltage. 10 = Select V_{REF} as reference voltage.
[15]	**TMTRGMOD**	**Timer Event Trigger ADC Conversion** 0 = This function Disabled. 1 = ADC Enabled by TiMER OUt event. setting TMSEL to select timer event from timer0~3
[14]	**Reserved**	Reserved.
[13:12]	**TMSEL**	**Select A/D Enable Time-out Source** 00 = TMR0. 01 = TMR1. 10 = TMR2. 11 = TMR3.

Bits	Description	
[11]	**ADST**	**A/D Conversion Start** 0 = Conversion stopped and A/D converter enter idle state. 1 = Conversion starts. ADST bit can be set to 1 from three sources: software write, external pin STADC and PWM trigger. ADST is cleared to 0 by hardware automatically at the end of single mode and single-cycle scan mode on specified channels. In continuous scan mode, A/D conversion is continuously performed sequentially unless software writes 0 to this bit or chip reset. **Note:** After ADC conversion done, SW needs to wait at least one ADC clock before to set this bit high again.
[10]	**DIFF**	**Differential Mode Selection** 0 = ADC is operated in single-ended mode. 1 = ADC is operated in differential mode. **Note:** Calibration should calibrated each time when switching between single-ended and differential mode
[9]	**PTEN**	**PDMA Transfer Enable Control** 0 = PDMA data transfer Disabled. 1 = PDMA data transfer in ADC_RESULT 0~17 Enabled. When A/D conversion is completed, the converted data is loaded into ADC_RESULT 0~10, software can enable this bit to generate a PDMA data transfer request. When PTEN=1, software must set ADIE=0 to disable interrupt. PDMA can access ADC_RESULT 0-17 registers by block or single transfer mode.
[8]	**TRGEN**	**External Trigger Enable Control** Enable or disable triggering of A/D conversion by external STADC pin. 0 = Disabled,. 1 = Enabled,.
[7:6]	**TRGCOND**	**External Trigger Condition** These two bits decide external pin STADC trigger event is level or edge. The signal must be kept at stable state at least 8 PCLKs for level trigger and 4 PCLKs at high and low state. 00 = Low level. 01 = High level. 10 = Falling edge. 11 = Rising edge.
[5:4]	**TRGS**	**Hardware Trigger Source** 00= A/D conversion is started by external STADC pin. 01= Reserved. 10= Reserved. 11= PWM trigger. Software should disable TRGE and ADST before change TRGS. In hardware trigger mode, the ADST bit is set by the external trigger from STADC or PWM trigger, However software has the highest priority to set or cleared ADST bit at any time.
[3:2]	**ADMD**	**A/D Converter Operation Mode** 00 = Single conversion. 01 = Reserved. 10 = Single-cycle scan. 11 = Continuous scan.

Bits	Description	
[1]	**ADIE**	**A/D Interrupt Enable Control** 0 = A/D interrupt function Disabled. 1 = A/D interrupt function Enabled. A/D conversion end interrupt request is generated if ADIE bit is set to 1.
[0]	**ADEN**	**A/D Converter Enable Control** 0 = Disabled. 1 = Enabled. Before starting A/D conversion, this bit should be set to 1. Clear it to 0 to disable A/D converter analog circuit power consumption.

A/D Channel Enable Register (ADCHER)

Register	Offset	R/W	Description	Reset Value
ADCHER	ADC_BA+0x4C	R/W	A/D Channel Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						CHEN17	CHEN16
15	14	13	12	11	10	9	8
CHEN15	CHEN14	Reserved					
7	6	5	4	3	2	1	0
CHEN7	CHEN6	CHEN5	CHEN4	CHEN3	CHEN2	CHEN1	CHEN0

Bits	Description	
[31:18]	Reserved	Reserved.
[17]	CHEN17	**Analog Input Channel 17 Enable Control (Convert AV_{SS})** 0 = Disabled. 1 = Enabled.
[16]	CHEN16	**Analog Input Channel 16 Enable Control (Convert AV_{DD})** 0 = Disabled. 1 = Enabled.
[15]	CHEN15	**Analog Input Channel 15 Enable Control (Convert Int_V_{REF})** 0 = Disabled. 1 = Enabled.
[14]	CHEN14	**Analog Input Channel 14 Enable Control (Convert V_{TEMP})** 0 = Disabled. 1 = Enabled.
[13:8]	Reserved	Reserved.
[7]	CHEN7	**Analog Input Channel 7 Enable Control (Convert Input Voltage From PA.7)** 0 = Disabled. 1 = Enabled.
[6]	CHEN6	**Analog Input Channel 6 Enable Control (Convert Input Voltage From PA.6)** 0 = Disabled. 1 = Enabled.
[5]	CHEN5	**Analog Input Channel 5 Enable Control (Convert Input Voltage From PA.5)** 0 = Disabled. 1 = Enabled.

Bits	Description	
[4]	**CHEN4**	**Analog Input Channel 4 Enable Control (Convert Input Voltage From PA.4)** 0 = Disabled. 1 = Enabled.
[3]	**CHEN3**	**Analog Input Channel 3 Enable Control (Convert Input Voltage From PA.3)** 0 = Disabled. 1 = Enabled.
[2]	**CHEN2**	**Analog Input Channel 2 Enable Control (Convert Input Voltage From PA.2)** 0 = Disabled. 1 = Enabled..
[1]	**CHEN1**	**Analog Input Channel 1 Enable Control (Convert Input Voltage From PA.1)** 0 = Disabled. 1 = Enabled.
[0]	**CHEN0**	**Analog Input Channel 0 Enable Control (Convert Input Voltage From PA.0)** 0 = Disabled. 1 = Enabled. If more than one channel in single mode is enabled by software, the least channel is converted and other enabled channels will be ignored.

A/D Compare Register 0/1 (ADCMPR0/1)

Register	Offset	R/W	Description	Reset Value
ADCMPR0	ADC_BA+0x50	R/W	A/D Compare Register 0	0x0000_0000
ADCMPR1	ADC_BA+0x54	R/W	A/D Compare Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				**CMPD**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CMPD							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**CMPMATCNT**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CMPCH					**CMPCOND**	**CMPIE**	**CMPEN**

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:16]	**CMPD**	**Comparison Data** The 12 bits data is used to compare with conversion result of specified channel. Software can use it to monitor the external analog input pin voltage variation in scan mode without imposing a load on software.
[15:12]	**Reserved**	Reserved.
[11:8]	**CMPMATCNT**	**Compare Match Count** When the specified A/D channel analog conversion result matches the compare condition defined by CMPCOND[2], the internal match counter will increase 1. When the internal counter reaches the value to (CMPMATCNT +1), the CMPF bit will be set.
[7:3]	**CMPCH**	**Compare Channel Selection** Set this field to select which channel’s result to be compared. **Note:** Valid setting of this field is channel 0~17, but channel 8~14 are reserved.
[2]	**CMPCOND**	**Compare Condition** 0 = Set the compare condition as that when a 12-bit A/D conversion result is less than the 12-bit CMPD (ADCMPRx[27:16]), the internal match counter will increase one. 1 = Set the compare condition as that when a 12-bit A/D conversion result is greater or equal to the 12-bit CMPD (ADCMPRx[27:16]), the internal match counter will increase by one. **Note:** When the internal counter reaches the value to (CMPMATCNT +1), the CMPF bit will be set.
[1]	**CMPIE**	**Compare Interrupt Enable Control** 0 = Compare function interrupt Disabled. 1 = Compare function interrupt Enabled. If the compare function is enabled and the compare condition matches the setting of CMPCOND and CMPMATCNT, CMPF bit will be asserted, in the meanwhile, if CMPIE is set to 1, a compare interrupt request is generated.

Bits	Description	
[0]	**CMPEN**	**Compare Enable Control** 0 = Compare Disabled. 1 = Compare Enabled. Set this bit to 1 to enable compare CMPD[11:0] with specified channel conversion result when converted data is loaded into ADC_RESULTx register. When this bit is set to 1, and CMPMATCNT is 0, the CMPF will be set once the match is hit.

A/D Status Register (ADSR)

Register	Offset	R/W	Description	Reset Value
ADSR	ADC_BA+0x58	R/W	A/D Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**INITRDY**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**CHANNEL**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CHANNEL				**BUSY**	**CMPF1**	**CMPF0**	**ADF**

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**INITRDY**	**ADC Power-up Sequence Completed** 0 = ADC not powered up after system reset. 1 = ADC has been powered up since the last system reset. **Note:** This bit will be set after system reset occurred and automatically cleared by power-up event.
[15:9]	**Reserved**	Reserved.
[8:4]	**CHANNEL**	**Current Conversion Channel (Read Only)** This filed reflects current conversion channel when BUSY=1. When BUSY=0, it shows the next channel to be converted.
[3]	**BUSY**	**BUSY/IDLE (Read Only)** 0 = A/D converter is in idle state. 1 = A/D converter is busy at conversion. **Note:** This bit is a mirror of ADST (ADCR[11]). That is to say if ADST(ADCR[11]) = 1,then BUSY is 1 and vice versa.
[2]	**CMPF1**	**Compare Flag** When the selected channel A/D conversion result meets setting condition in ADCMPR1 then this bit is set to 1. And it is cleared by writing 1 to self. 0 = Conversion result in ADC_RESULTx does not meet ADCMPR1 setting. 1 = Conversion result in ADC_RESULTx meets ADCMPR1 setting. This flag can be cleared by writing 1 to it. **Note:** When this flag is set, the matching counter will be reset to 0,and continue to count when user writes 1 to clear CMPF1.

Bits	Description	
[1]	**CMPF0**	**Compare Flag** When the selected channel A/D conversion result meets setting condition in ADCMPR0 then this bit is set to 1. And it is cleared by writing 1 to self. 0 = Conversion result in ADC_RESULTx does not meet ADCMPR0setting. 1 = Conversion result in ADC_RESULTx meets ADCMPR0setting. This flag can be cleared by writing 1 to it. **Note:** When this flag is set, the matching counter will be reset to 0,and continue to count when user write 1 to clear CMPF0
[0]	**ADF**	**A/D Conversion End Flag** A status flag that indicates the end of A/D conversion. ADF is set to 1 at these two conditions: When A/D conversion ends in single mode When A/D conversion ends on all specified channels in scan mode. This flag can be cleared by writing 1 to it.

A/D PDMA Current Transfer Data Register (ADPDMA)

Register	Offset	R/W	Description	Reset Value
ADPDMA	ADC_BA+0x60	R	A/D PDMA Current Transfer Data Register	0x0000_0000

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th></tr>
<tr><td colspan="4">Reserved</td><td colspan="4">AD_PDMA</td></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">AD_PDMA</td></tr>
</table>

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11:0]	**AD_PDMA**	**ADC PDMA Current Transfer Data (Read Only)** When PDMA transferring, reading these bits can monitor the current PDMA transfer data.

A/D Power Management Register (ADCPWD)

Register	Offset	R/W	Description	Reset Value
ADCPWD	ADC_BA+0x64	R/W	ADC Power Management Register	0x0000_0A00

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						**Reserved**	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**PWDMOD**		**PWDCALEN**	**PWUPRDY**

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3:2]	**PWDMOD**	**ADC Power-down Mode** Set this bit fields to select ADC Power-down mode when system power-down. 00 = Reserved. 01 = ADC Power-down mode. 10 = ADC Standby mode. 11 = Reserved. **Note1:** Different PWDMOD has different power down/up sequence, in order to avoid ADC powering up with wrong sequence; user must keep PWMOD consistent each time in power down and power up. **Note2:** While the ADC is power up from Power-down mode without calibration, the PWDCALEN(ADCPWD[1]) is set to 0. (The calibration value will be reset)
[1]	**PWDCALEN**	**Power Up Calibration Function Enable Control** 0 = Power up without calibration. 1 = Power up with calibration. **Note:** This bit work together with CALSEL (ADCCALCTL[3]), see the following {PWDCALEN,CALFBSEL} Description: PWDCALEN is 0 and CALFBSEL is 0: No need to calibrate. PWDCALEN is 0 and CALFBSEL is 1: No need to calibrate. PWDCALEN is 1 and CALFBSEL is 0: Load calibration word when power up. PWDCALEN is 1 and CALFBSEL is 1: Calibrate when power up.
[0]	**PWUPRDY**	**ADC Power-up Sequence Completed and Ready for Conversion** 0 = ADC is not ready for conversion may be in power down state or in the progress of power up. 1 = ADC is ready for conversion.

A/D calibration control register (ADCCALCTL)

Register	Offset	R/W	Description	Reset Value
ADCCALCTL	ADC_BA+0x68	R/W	ADC Calibration Control Register	0x0000_0009

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				CALSEL	CALDONE	CALSTART	CALEN

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**CALSEL**	**Select Calibration Functional Block** 0 = Load calibration functional block. 1 = Calibration functional block.
[2]	**CALDONE**	**Calibrate Functional Block Complete** 0 = Not yet. 1 = Selected functional block complete.
[1]	**CALSTART**	**Calibration Functional Block Start** 0 = Stops calibration functional block. 1 = Starts calibration functional block. **Note:** This bit is set by SW and clear by HW; don't write 1 to this bit while CALEN = 0.
[0]	**CALEN**	**Calibration Function Enable Control** Enable this bit to turn on the calibration function block. 0 = (BYPASSCAL). 1 = Enabled.

A/D Calibration Word (ADCCALWORD)

Register	Offset	R/W	Description	Reset Value
ADCCALWORD	ADC_BA+0x6C	R/W	A/D calibration load word register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**CALWORD**						

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6:0]	**CALWORD**	**Calibration Word Bits** Write to this register with the previous calibration word before load calibration action Read this register after calibration done **Note:** The calibration block contains two parts “CALIBRATION“ and “LOAD CALIBRATION”; if the calibration block configure as “CALIBRATION“; then this register represent the result of calibration when calibration is completed; if configure as “LOAD CALIBRATION” ; configure this register before loading calibration action, after loading calibration complete, the laoded calibration word will apply to the ADC; while in loading calibration function the loaded value will not be equal to the original CALWORD until calibration is done.

A/D Channel Samping0 Register (ADCCHSAMP0)

Register	Offset	R/W	Description	Reset Value
ADCCHSAMP0	ADC_BA+0x70	R/W	ADC Channel Sampling Time Counter Register Group 0	0x0000_0000

31	30	29	28	27	26	25	24
CH7SAMPCNT				CH6SAMPCNT			
23	22	21	20	19	18	17	16
CH5SAMPCNT				CH4SAMPCNT			
15	14	13	12	11	10	9	8
CH3SAMPCNT				CH2SAMPCNT			
7	6	5	4	3	2	1	0
CH1SAMPCNT				CH0SAMPCNT			

Bits	Description	
[31:28]	**CH7SAMPCNT**	**Channel 7 Sampling Counter** The same as Channel 0 sampling counter table.
[27:24]	**CH6SAMPCNT**	**Channel 6 Sampling Counter** The same as Channel 0 sampling counter table.
[23:20]	**CH5SAMPCNT**	**Channel 5 Sampling Counter** The same as Channel 0 sampling counter table.
[19:16]	**CH4SAMPCNT**	**Channel 4 Sampling Counter** The same as Channel 0 sampling counter table.
[15:12]	**CH3SAMPCNT**	**Channel 3 Sampling Counter** The same as Channel 0 sampling counter table.
[11:8]	**CH2SAMPCNT**	**Channel 2 Sampling Counter** The same as Channel 0 sampling counter table.
[7:4]	**CH1SAMPCNT**	**Channel 1 Sampling Counter** The same as Channel 0 sampling counter table.

Bits	Description	
[3:0]	**CH0SAMPCNT**	**Channel 0 Sampling Counter** CH0SAMPCNT ADC Clock 0 = 0. 1 = 1. 2 = 2. 3 = 4. 4 = 8. 5 = 16. 6 = 32. 7 = 64. 8 = 128. 9 = 256. 10 = 512. 11 = 1024. 12 = 1024. 13 = 1024. 14 = 1024. 15 = 1024.

A/D Channel Samping1 Register (ADCCHSAMP1)

Register	Offset	R/W	Description	Reset Value
ADCCHSAMP1	ADC_BA+0x74	R/W	ADC Channel Sampling Time Counter Register Group 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				**INTCHSAMPCNT**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19:16]	**INTCHSAMPCNT**	**Internal Channel (V_{TEMP}, AV_{DD}, AV_{SS}, Int_V_{REF}) Sampling Counter** The same as Channel 0 sampling counter table.
[15:0]	**Reserved**	Reserved.

6.20 Analog Comparator Controller (ACMP)

6.20.1 Overview

The Nano112 series contains two comparators. The comparator output is logic 1 when positive input is greater than negative input; otherwise, the output is 0. Each comparator can be configured to generate an interrupt when the comparator output value changes. The comparator ACMP0 can be used as normal comparator or it can emulate ADC function. The comparator ACMP1 can be used as normal comparator only.

6.20.2 Features

- Analog input voltage range: 0 ~ AV_{DD}
- Supports hysteresis function
- Supports wake-up function
- Comparator ACMP0 supports
  - 4 positive sources(ACMP0_Px)
    - PA.1, PA.2, PA.3, or PA.4
  - 4 negative sources
    - PA.5 (ACMP0_N)
    - Comparator Reference Voltage (CRV)
    - Int_V_{REF}
    - AGND
- Comparator ACMP1 supports
  - 1 positive source
    - PA.12(ACMP1_P)
  - 4 negative sources
    - PA.13(ACMP1_N)
    - Comparator Reference Voltage (CRV)
    - Int_V_{REF}
    - AGND
- Comparator ACMP0 supports three operation modes:
  - Normal Comparator mode
  - Single Slope ADC mode: Resistance measurement (e.g. PTC, NTC, PT1000)
    - Supports to measure 7 channels resistor
  - Sigma-Delta ADC mode
    - Supports up to 4 channel voltage input from ACMP0_Px

6.20.3 Block Diagram

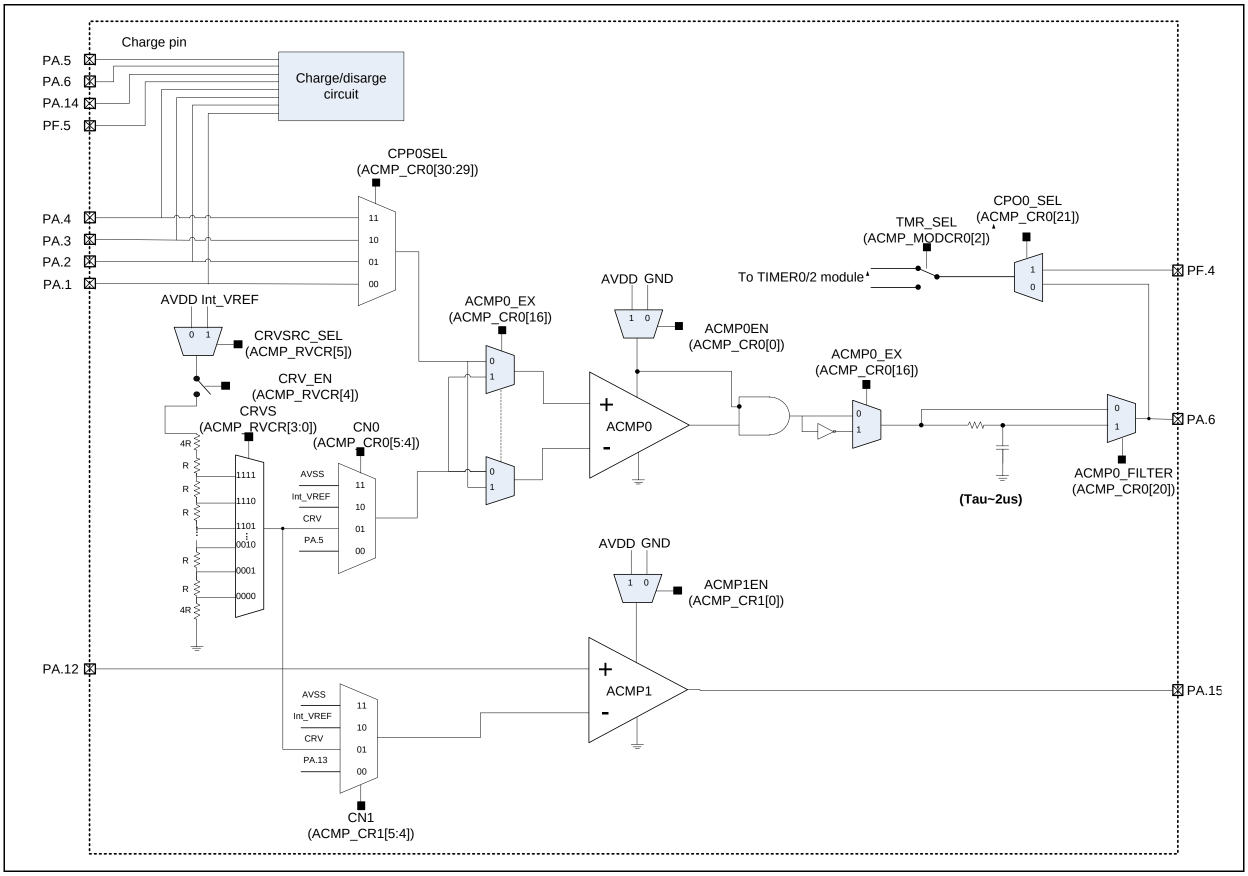


Figure 6-131 Analog Comparator Block Diagram

6.20.4 Functional Description

6.20.4.1 Comparator Reference Voltage (CRV)

The comparator reference voltage (CRV) module is responsible for generating reference voltage for comparators. The CRV module consists of resisters ladder and analog switch, and user can set the CRV output voltage using the CRVS(ACMP_RVCR[3:0]) register and select the reference voltage to ACMP by setting the CRVSRC_SEL(ACMP_RVCR[5]) register.

Features:

1. User selectable references voltage source by setting the CRVSRC_SEL(ACMP_RVCR[5]) register.
2. User selectable references voltage by setting the CRVS(ACMP_RVCR[3:0]) register.

Comparator reference voltage = V_{IN} * (1/6+CRVS[3:0]/24); V_{IN} = AV_{DD} or Int_V_{REF}

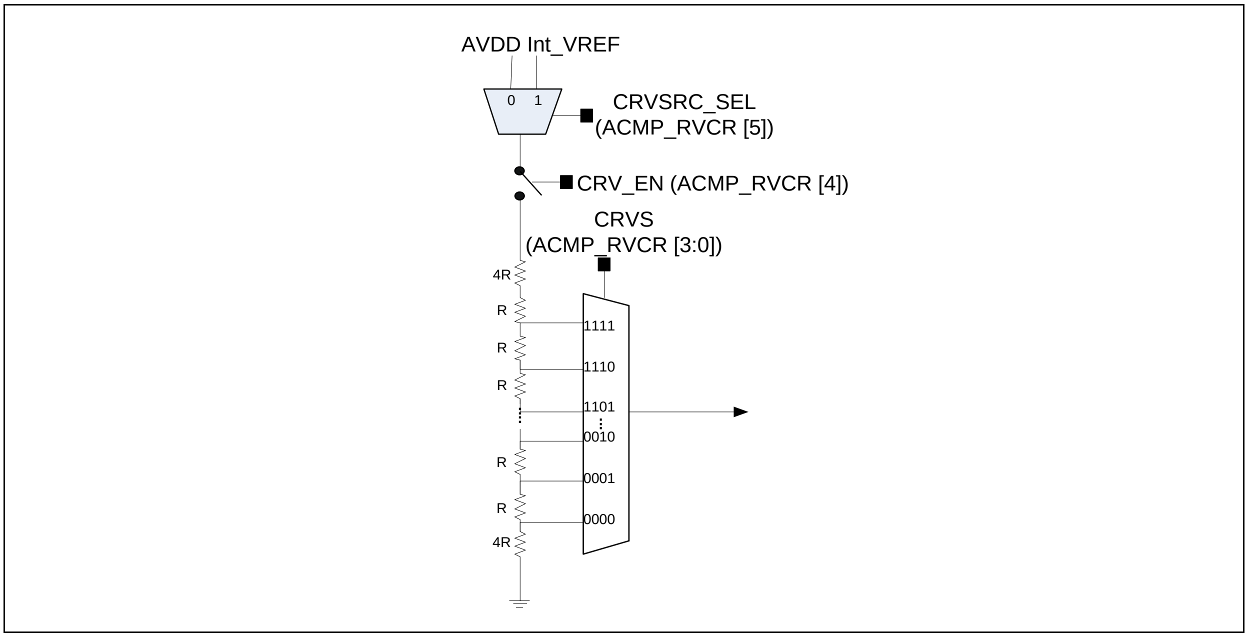


Figure 6-132 Comparator Reference Voltage Block Diagram

6.20.4.2 Normal Comparator Mode

Both the comparator ACMP0 and comparator ACMP1 have normal comparator mode. The following shows the general application circuit for the normal mode.

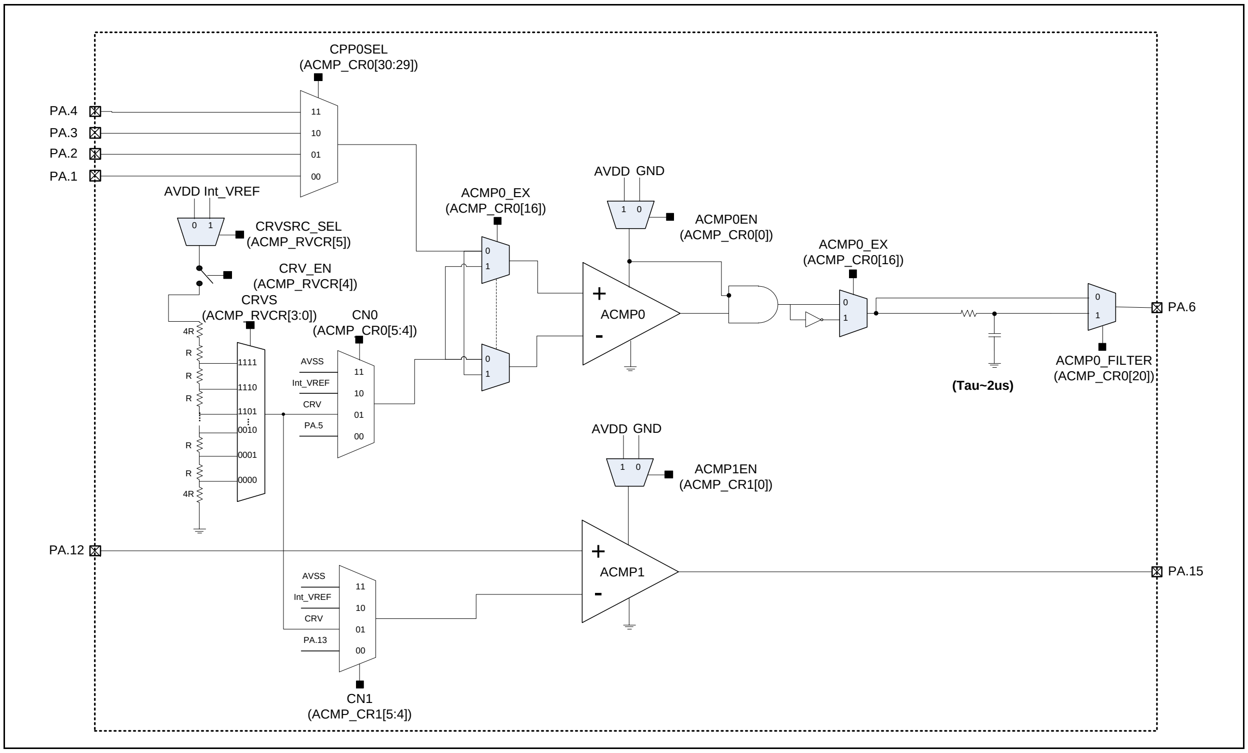


Figure 6-133 Normal Comparator Mode Block Diagram

6.20.4.3 *ACMP0 Single Slope ADC Mode*

The Single Slop ADC mode is an analog-to-digital conversion technique to measure unknown resistance and can be implemented with a comparator rather than a standalone ADC module. The technique is using known and unknown resistor to discharge a capacitor. While the capacitor is discharged, the number of clock cycles counted by timer indicates resistance ratio. Larger counting number indicates Larger resistance value. The unknown resistance can be calculated based on the discharge time and known resistance.

Only the comparator ACMP0 has Single Slope ADC mode which can implement a low-cost but high resolution ADC function by using just a little additional external components. The external components include a thermistor (e.g. PT1000), a reference resistor, and a capacitor. The components are connected directly to the Nano112 as shown in Figure 6‑134.

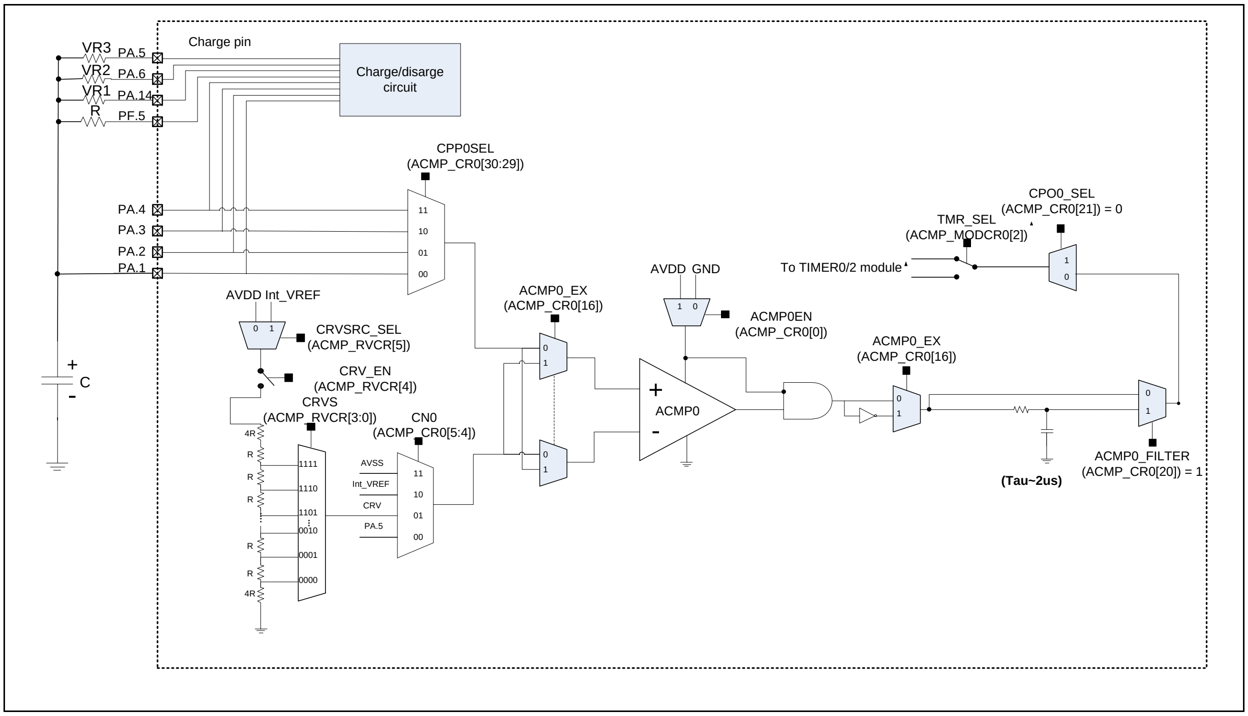


Figure 6‑134 ACMP Single Slope ADC Mode Reference Circuit

Both of TIMER0 and TIMER2 can support the Single Slope ADC mode. Below description uses TIMER0 as an example.

ACMP0 controller is set as single slope convert with following steps:

1. Set ACMP_EN_TMR (TMR0_CTL[6]) to 1, allow ACMP0 can enable TIMER0.
2. Select using TIMER0, refer to TMR_SEL (ACMP_MODCR0[2]).
3. Select ACMP0_Px, refer to CPP0SEL(ACMP_CR0[30:29]).
4. Set TMR_TRI_LV (ACMP_MODCR0[3]) to 0, let’s comparator output from low to high to trigger the Timer.
5. Set MOD_SEL (ACMP_MODCR0[1:0]) to 2 (Single Slope mode).
6. Select charge/discharge pin, refer to CH_DIS_FUN_SEL (ACMP_MODCR0[6:4]).
7. Wait capacitor saturation.

8. Set START (ACMP_MODCR0[8]) to 1, to start Single Slope convert.
9. Wait TIMER0 interrupt then check Timer capture counter.
10. Set START (ACMP_MODCR0[8]) to 0.

For next Single Slope conversion, repeat Step 6 to 10.

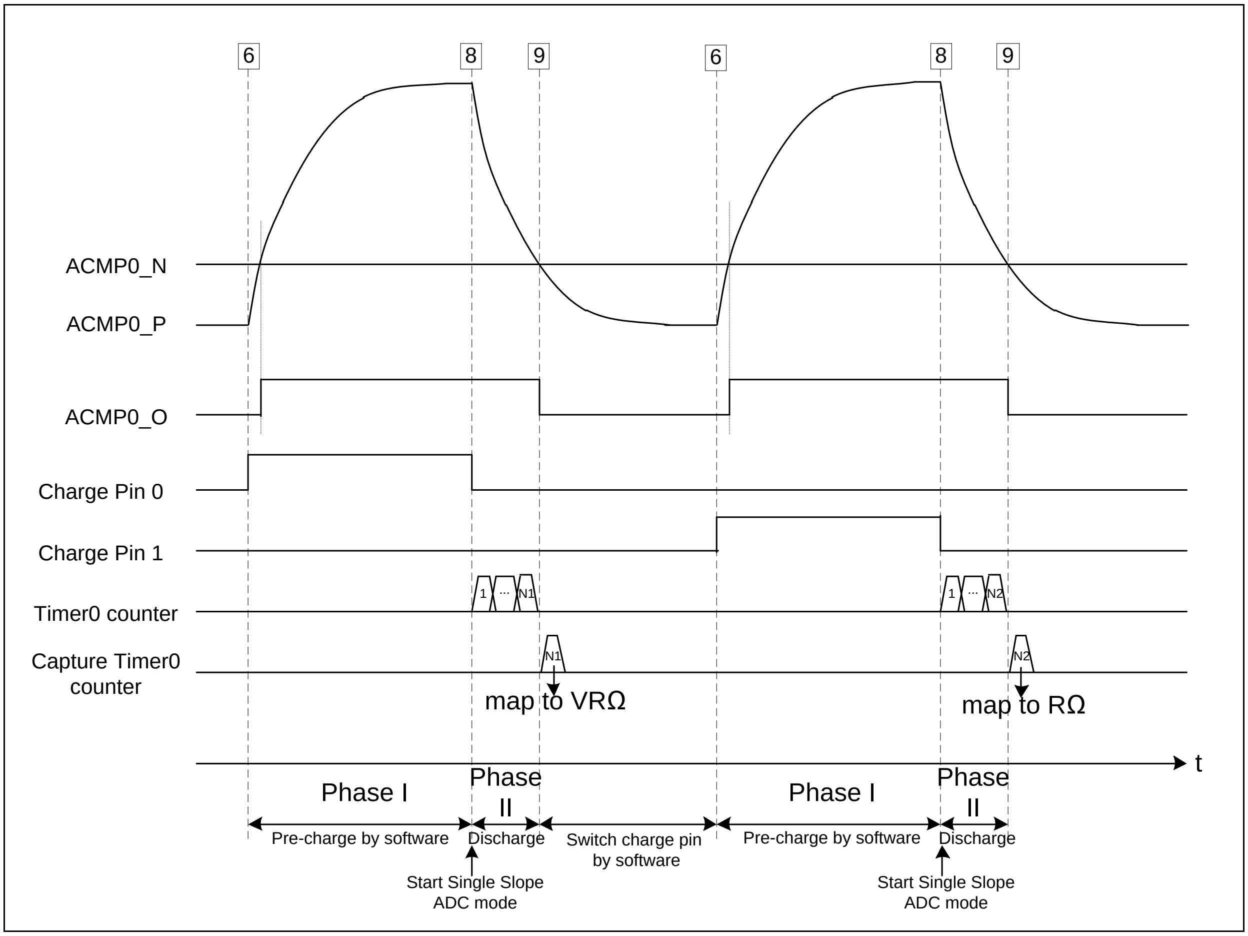


Figure 6-135 ACMP0 Single Slope ADC Mode Timing

- **Phase I**: The software will pre-charge (refer to single slope convert step 6) on C(Capacitor) until saturation.
- **Phase II**: After setting the programming flow, hardware will start TIMER0 and discharge the capacitor at the same time. Till ACMP0_O reverses, TIMER0 counter stops and the value can be calculated as discharge time.

6.20.4.4 *ACMP0 Sigma-Delta ADC Mode*

The Sigma-Delta principle is becoming more and more important for high-resolution ADCs and is proven in many applications. The concept of an integrating A/D converter is to match an unknown voltage of interest ACMP0_Px, with a known voltage, V_{DD}.

Only the comparator ACMP0 has Sigma-Delta ADC mode which can implement a low-cost but high resolution ADC function by using just a little additional external components. The external components include a resistor and a capacitor. The components are connected directly to the Nano112 as shown in the following figure.

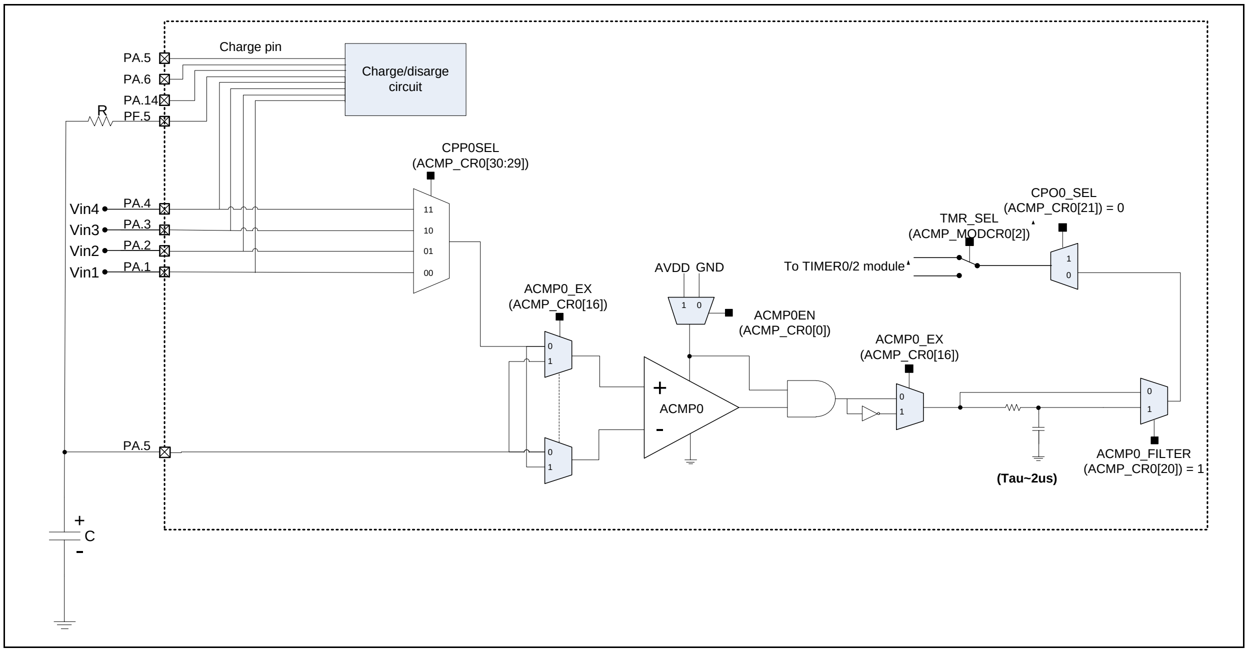


Figure 6-136 ACMP Sigma-Delta ADC Mode Reference Circuit

In Sigma Delta ADC mode, ACMP0 would output high on the charge pin to charge the capacitor first. After analog comparator output is inversed, ACMP0 will change to output low on the charge pin to dis-charge the capacitor. Similarly, after analog comparator output is inversed again, ACMP0 will change to output high on charge pin to charge the capacitor again.

Sigma Delta ADC mode needs ACMP0 and two Timers. Both of TIMER0/TIMER1 and TIMER2/TIMER3 can support the Sigma Delta ADC mode. Below description uses TIMER0/TIMER1 as an example.

The number of charge and dis-charge process is defined in TMR_CMP (TMR0_CMPR[23:0]). If setting 1500 to TMR_CMP (TMR0_CMPR[23:0]), the charge and dis-charge process would go for 3000 times.

The software setting for sigma-delta convert is described with following steps:

1. Set MOD_SEL (ACMP_MODCR0[1:0]) to 1 to select Sigma-Delta ADC mode.
2. Set TIMER0 as One-shot mode for Sigma-Delta ADC mode conversion clock.
3. Set ACMP_EN_TMR (TMR0_CTL[6]) to 1 that allows ACMP0 can enable TIMER0.
4. Set EVENT_EN (TMR1_CTL[12]) to 1 that enables TIMER1 Event Counting mode to get Sigma-Delta ADC mode conversion data.
5. Set ACOMP0_PN_AutoEx (ACMP_CR0[19]) to 1 that enables ACMP0 to swap the positive with negative input and inverse ACMP0 output level while TIMER0 counter overflow. The swapping can cancel the offset error of the comparator.

6. Set TMR_SEL (ACMP_MODCR0[2]) to 0 to use TIMER0 and TIMER1.
7. Set CH_DIS_FUN_SEL (ACMP_MODCR0[6:4]) to select the charge/discharge pin.
8. Set TMR_TRI_LV (ACMP_MODCR0[3]) to 1 that enables the TIMER0 while ACMP0 output changes from high to low.
9. Set CPP0SEL (ACMP_CR0[30:29]) to select the input pin of ACMP0_Px, which is the voltage to be measured by Sigma-Delta ADC.
10. Set START (ACMP_MODCR0[8]) to 1 to start Sigma-Delta ADC conversion.
11. Wait TIMER0 interrupt and then read the TIMER1 capture data register (TMR1_TCAP) for ADC conversion data.
12. Set START (ACMP_MODCR0[8]) to 0 for correcting the initial condition of next conversion.

For the next Sigma-Delta conversion, repeat Step 9 to 12.

Refer to following diagram for detailed conversion timing.

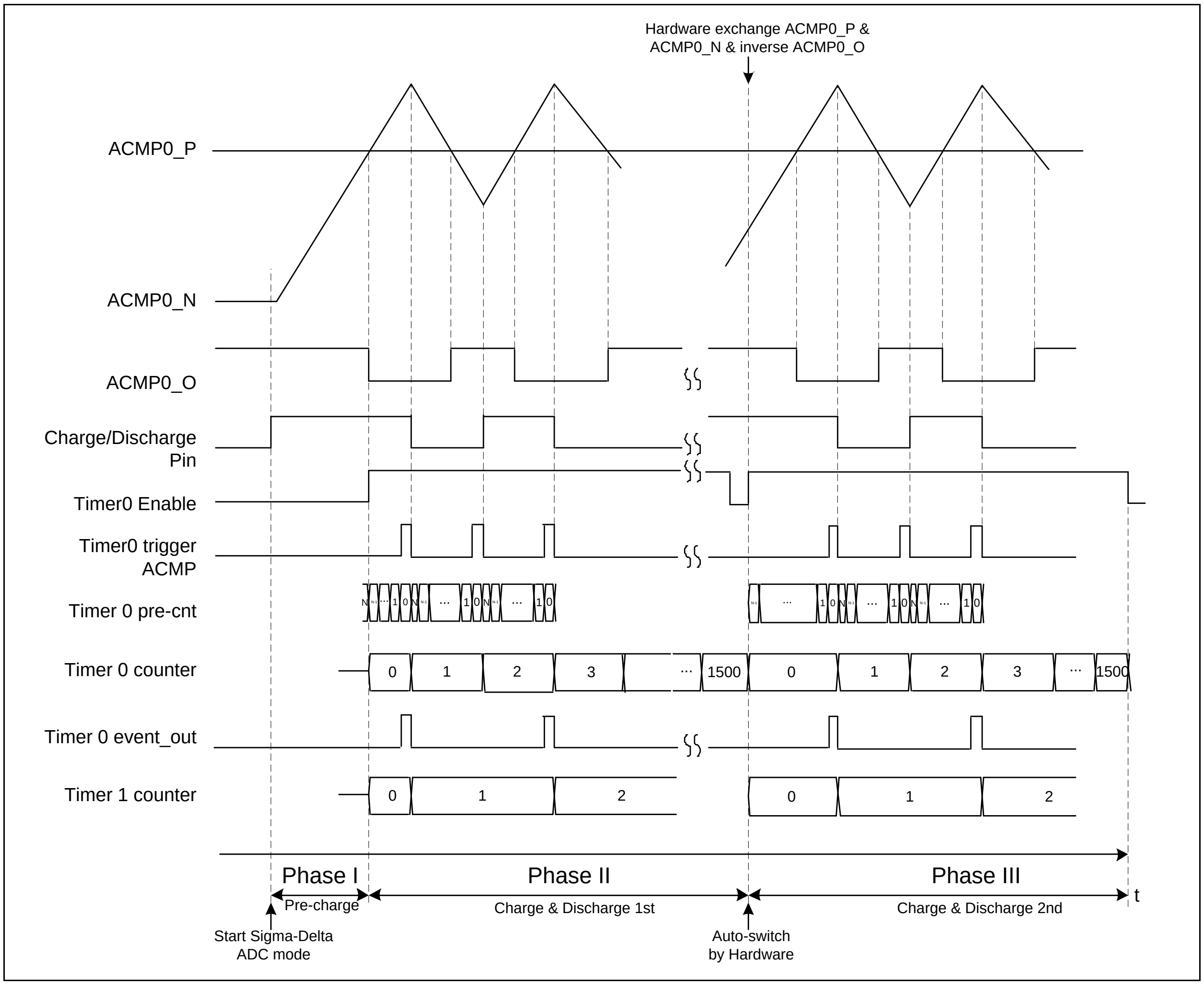


Figure 6-137 ACMP Sigma-Delta ADC Mode Timing

After setting the software according to programming flow (refer to sigma-delta convert steps), hardware behaves from **Phase I** to **III**:

- **Phase I**: Drive "charge/discharge pin" high to charge capacitor (C in Figure 6‑136) until the ACMP0_O reverses and then start TIMER0.
- **Phase II**: Continue to charge or discharge capacitor depending on ACMP0_O value. If ACMP0_O = 0, drive "charge/discharge pin" low to discharge capacitor. Otherwise, drive "charge/discharge pin" high to charge capacitor. Simultaneously, TIMER1 counter value plus 1 If ACMP0_O = 0.
  - In the end of Phase II: TIMER0 counter overflow will swap ACMP0_Px with ACMP0_N and inverse ACMP0_O.
- **Phase III**: Do the same flow as **Phase II**.

Software reads TIMER1 event counter value after completing the above procedures. This value could be calculated to get ACMP0_Px voltage. The formula is listed as below:

$$V_{IN} = \frac{\mathrm{TMR1\_TCAP}}{TMR0\_CMPR \times 2} \times V_{DD}$$

V_{IN}: the voltage of ACMP0_Px

For example, if AV_{DD} is 3000mV, the voltage of "charge/discharge pin" is 3000mV, too. For the convenience of calculation, TMR0_CMPR can be set to 15000. After conversion is finished, if TMR1_TCAP is 18000, the voltage of ACMP0_Px (V_{IN}) is calculated as 1800mV.

6.20.4.5 Interrupt Sources

The comparator generates an output CO0 (ACMP_SR[2]) and CO1 (ACMP_SR[3]) in ACMP_SR register which is sampled by PCLK. If ACMP0IE (ACMP_CR0[1]) and ACMP1IE (ACMP_CR1[1]) bit in ACMP_CR0 and ACMP_CR1 are set, a state change on the comparator output CO0 (ACMP_SR[2]) and CO1 (ACMP_SR[3]) will cause comparator flag ACMPF0 (ACMP_SR[0]) and ACMPF1 (ACMP_SR[1]) set and the comparator interrupt requested. Software can write 1 to ACMPF0 (ACMP_SR[0]) and ACMPF1 (ACMP_SR[1]) to stop interrupt request.

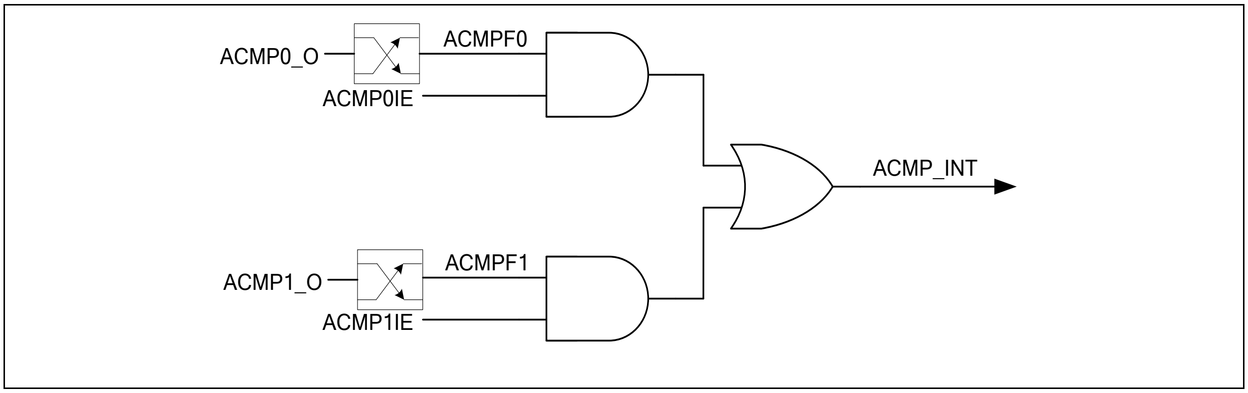


Figure 6‑138 Analog Comparator Interrupt Sources

6.20.5 Register Map

R: read only, **W**: write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
ACMP Base Address: **ACMP_BA = 0x401D_0000**				
ACMP_CR0	ACMP_BA+0x00	R/W	Analog Comparator 0 Control Register	0x0000_0000
ACMP_CR1	ACMP_BA+0x04	R/W	Analog Comparator 1 Control Register	0x0000_0000
ACMP_SR	ACMP_BA+0x08	R/W	Analog Comparator Status Register	0x0000_0000
ACMP_RVCR	ACMP_BA+0x0C	R/W	Analog Comparator Reference Voltage Control Register	0x0000_0000
ACMP_MODCR0	ACMP_BA+0x10	R/W	Analog Comparator 0 Mode Control Register	0x0000_0000

6.20.6 Register Description

Analog Comparator 0 Control Register (ACMP_CR0)

Register	Offset	R/W	Description	Reset Value
ACMP_CR0	ACMP_BA+0x00	R/W	Analog Comparator 0 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
ACMP0_WKEUP_EN	**CPP0SEL**		**Reserved**				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**CPO0_SEL**	**ACMP0_FILTER**	**ACOMP0_PN_AutoEx**	**Reserved**		**ACMP0_EX**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**CN0**		**Reserved**	**ACMP0_HYSEN**	**ACMP0IE**	**ACMP0EN**

Bits	Description	
[31]	**ACMP0_WKEUP_EN**	**Comparator ACMP0 Wake-up Enable Control** 0 = Wake-up function Disabled**.** 1 = Wake-up function Enabled when the system enters Power-down mode.
[30:29]	**CPP0SEL[1:0]**	**Comparator ACMP0 Positive Input Selection** 00 = Input from PA.4. 01 = Input from PA.3. 10 = Input from PA.2. 11 = Input from PA.1.
[28:22]	**Reserved**	Reserved.
[21]	**CPO0_SEL**	**Comparator ACMP0 Output to Timer Path Selection** 0 = Comparator ACMP0 output to Timer is through internal path. 1 = Comparator ACMP0 output to Timer is through external pin (through PF.4).
[20]	**ACMP0_FILTER**	**Comparator ACMP0 Output Filter** 0 = Comparator ACMP0 output is not filtered by internal RC filter. 1 = Comparator ACMP0 output is filtered by internal RC filter.
[19]	**ACOMP0_PN_AutoEx**	**Comparator Analog ACMP0_Px & ACMP0_N Input Swap Function Automatically** This bit is only for sigma-delta ADC mode use. 0 = Disabled to swap comparator ACMP0 input function, ACMP0_Px and ACMP0_N, automatically. 1 = Enabled to swap comparator ACMP0 input function, ACMP0_Px and ACMP0_N, automatically.
[18:17]	**Reserved**	Reserved.

Bits	Description	
[16]	**ACMP0_EX**	**Comparator ACMP0 Swap** 0 = No swap to the comparator inputs and output. 1 = Swap the comparator inputs with ACMP0_Px and ACMP0_N, and invert the polarity of comparator 0 output. **Note:** This bit swaps the comparator inputs and inverts the comparator output.
[15:6]	**Reserved**	Reserved.
[5:4]	**CN0**	**Comparator ACMP0 Negative Input Selection** 00 = The comparator reference pin ACMP0_N is selected as the negative comparator input. 01 = The internal comparator reference voltage (CRV) is selected as the negative comparator input. 10 = The internal reference voltage (Int_V_{REF}) is selected as the negative comparator input. 11 = The AGND is selected as the negative comparator input.
[3]	**Reserved**	Reserved.
[2]	**ACMP0_HYSEN**	**Comparator ACMP0 Hysteresis Enable Control** 0 = ACMP0 Hysteresis function Disabled. 1 = ACMP0 Hysteresis function Enabled. The typical range is 20mV.
[1]	**ACMP0IE**	**Comparator ACMP0 Interrupt Enable Control** 0 = ACMP0 interrupt function Disabled. 1 = ACMP0 interrupt function Enabled. **Note:** Interrupt generated if ACMP0IE bit is set to “1” after ACMP0 output changed.
[0]	**ACMP0EN**	**Comparator ACMP0 Enable Control** 0 = Disabled. 1 = Enabled. **Note:** Comparator output needs to wait 10 μs stable time after ACMP0EN is set.

Analog Comparator 1 Control Register (ACMP_CR1)

Register	Offset	R/W	Description	Reset Value
ACMP_CR1	ACMP_BA+0x04	R/W	Analog Comparator 1 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
ACMP1_WKEUP_EN	**Reserved**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**CN1**		**Reserved**	**ACMP1_HYSEN**	**ACMP1IE**	**ACMP1EN**

Bits	Description	
[31]	**ACMP1_WKEUP_EN**	**Comparator ACMP1 Wake-up Enable Control** 0 = Wake-up function Disabled. 1 = Wake-up function Enabled when the system enters Power-down mode.
[30:6]	**Reserved**	Reserved.
[5:4]	**CN1**	**Comparator ACMP1 Negative Input Selection** 00 = The comparator reference pin ACMP0_N is selected as the negative comparator input. 01 = The internal comparator reference voltage (CRV) is selected as the negative comparator input. 10 = The internal reference voltage (Int_V_{REF}) is selected as the negative comparator input. 11 = The AGND is selected as the negative comparator input.
[3]	**Reserved**	Reserved.
[2]	**ACMP1_HYSEN**	**Comparator ACMP1 Hysteresis Enable Control** 0 = ACMP1 Hysteresis function Disabled. 1 = ACMP1 Hysteresis function Enabled. The typical range is 20mV.
[1]	**ACMP1IE**	**Comparator ACMP1 Interrupt Enable Control** 0 = ACMP1 interrupt function Disabled. 1 = ACMP1 interrupt function Enabled. **Note:** Interrupt is generated if ACMP0IE bit is set to “1” after ACMP1 output changed.
[0]	**ACMP1EN**	**Comparator ACMP1 Enable Control** 0 = Disabled. 1 = Enabled. **Note:** Comparator output needs to wait 10 µs stable time after ACMP1EN is set.

Analog Comparator Status Register (ACMP_SR)

Register	Offset	R/W	Description	Reset Value
ACMP_SR	ACMP_BA+0x08	R/W	Analog Comparator Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**CO1**	**CO0**	**ACMPF1**	**ACMPF0**

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**CO1**	**Comparator ACMP1 Output** Synchronized to the PCLK to allow reading by software. Cleared when the comparator is disabled (ACMP1EN = 0).
[2]	**CO0**	**Comparator ACMP0 Output** Synchronized to the PCLK to allow reading by software. Cleared when the comparator is disabled (ACMP0EN = 0).
[1]	**ACMPF1**	**Comparator ACMP1 Flag** This bit is set by hardware whenever the comparator 1 output changes state. This will generate an interrupt if ACMP1IE set. **Note:** Write “1” to clear this bit to 0.
[0]	**ACMPF0**	**Comparator ACMP0 Flag** This bit is set by hardware whenever the comparator 0 output changes state. This will generate an interrupt if ACMP0IE set. **Note:** Write “1” to clear this bit to 0.

Analog Comparator Reference Voltage Control Register (ACMP_RVCR)

Register	Offset	R/W	Description	Reset Value
ACMP_RVCR	ACMP_BA+0x0C	R/W	Analog Comparator Reference Voltage Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**CRVSRC_SEL**	**CRV_EN**	**CRVS**			

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**CRVSRC_SEL**	**CRV Source Selection** 0 = From AV_{DD}. 1 = From Int_V_{REF}.
[4]	**CRV_EN**	**CRV Enable Control** 0 = CRV Disabled. 1 = CRV Enabled.
[3:0]	**CRVS[3:0]**	**Comparator Reference Voltage Setting** Comparator reference voltage = V_{IN} * (1/6+CRVS[3:0]/24). V_{IN} = AV_{DD} or Int_$V_{REF.}$

NANO102/112 SERIES TECHNICAL REFERENCE MANUAL

Analog Comparator 0 Mode Control Register (ACMP_MODCR0)

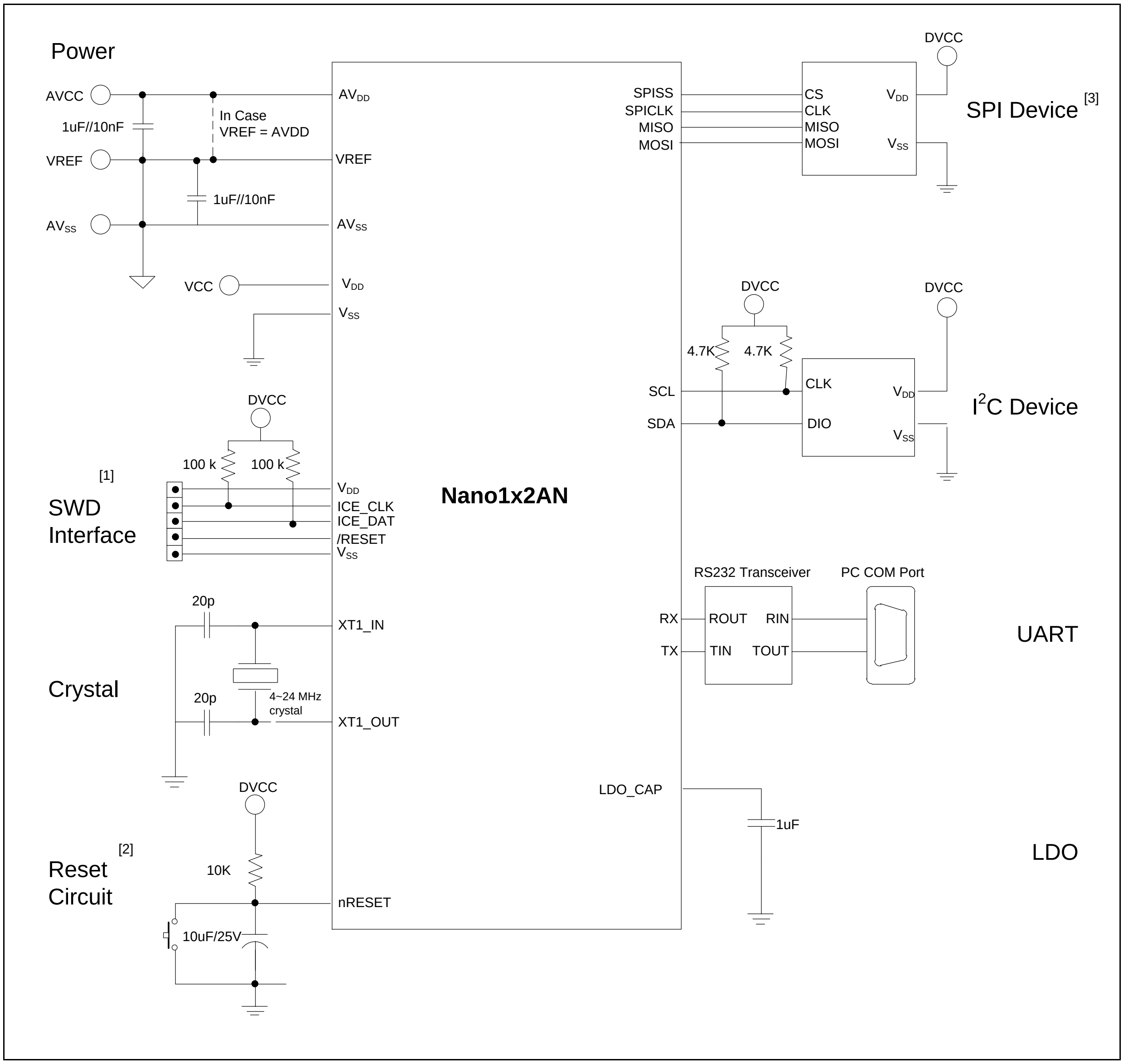
Register	Offset	R/W	Description	Reset Value
ACMP_MODCR0	ACMP_BA+0x10	R/W	Analog Comparator 0 Mode Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**START**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CH_DIS_FUN_SEL	**CH_DIS_PIN_SEL**			**TMR_TRI_LV**	**TMR_SEL**	**MOD_SEL**	

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**START**	**Start ADC Mode** 0 = Stop Sigma-Delta ADC Mode or Single Slope ADC Mode. 1 = Start Sigma-Delta ADC Mode or Single Slope ADC Mode.
[7]	**CH_DIS_FUN_SEL**	**Charge or Discharge Pin Function Option** This bit is for Single Slope ADC Mode only. 0 = Drive low on charge pin to dis-charge capacitor and drive high on charge pin to charge capacitor. 1 = Drive high on charge pin to dis-charge capacitor and drive low on charge pin to charge capacitor.
[6:4]	**CH_DIS_PIN_SEL**	**Charge or Discharge Pin Selection** 000 = PA.1. 001 = PA.2. 010 = PA.3. 011 = PA.4. 100 = PA.5. 101 = PA.6. 110 = PA.14. 111 = PF.5.
[3]	**TMR_TRI_LV**	**Timer Trigger Level** This bit is for Sigma-Delta ADC Mode. 0 = Comparator Output Low to High to Enable Timer. 1 = Comparator Output High to Low to Enable Timer.
[2]	**TMR_SEL**	**Analog Comparator 0 Co-operation Timer Selection** 0 = Select TIMER0 as co-operation Timer. 1 = Select TIMER2 as co-operation Timer.

Bits	Description	
[1:0]	**MOD_SEL**	**Comparator Mode Selection** 00 = Normal Comparator Mode. 01 = Sigma-Delta ADC Mode. 10 = Single Slope ADC Mode. 11 = Reserved.

7 APPLICATION CIRCUIT



Note 1: It is recommended to use 100 kΩ pull-up resistor on both ICE_DAT and ICE_CLK pin.

Note 2: It is recommended to use 10 kΩ pull-up resistor and 10 µF capacitor on nRESET pin.

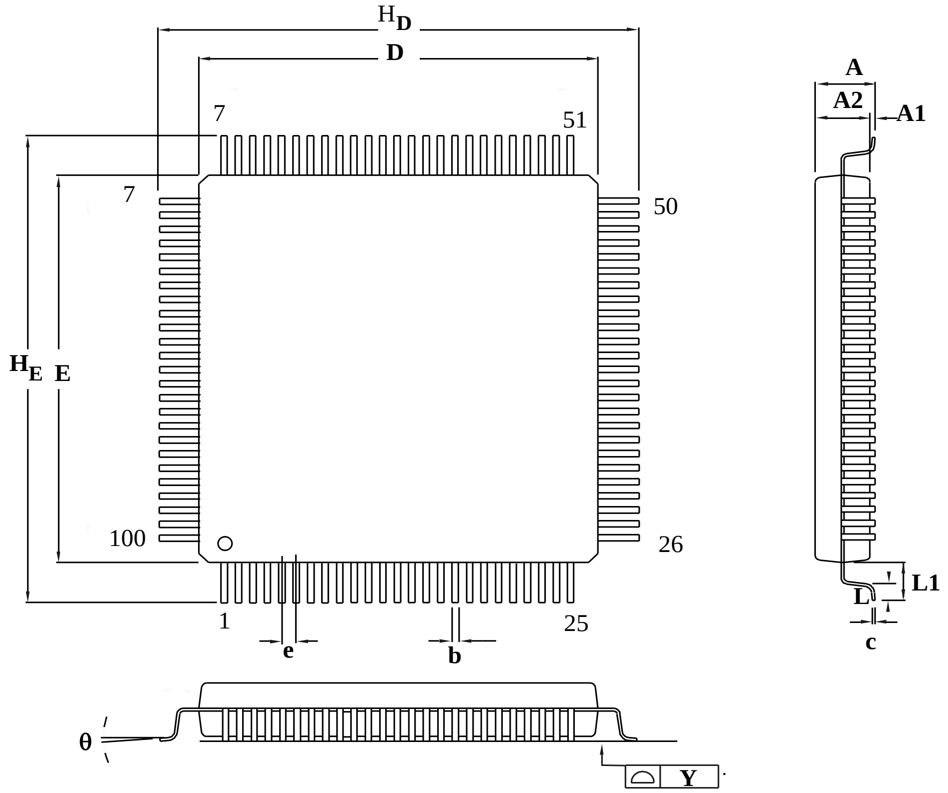
Note 3: For the SPI device, the chip supply voltage must be equal to SPI device working voltage. For example, when the SPI Flash working voltage is 3.3 V, the Nano1x2 chip supply voltage must also be 3.3 V

8 ELECTRICAL CHARACTERISTICS

For information on the Nano102/Nano112 series electrical characteristics, please refer to NuMicro® Nano102/Nano112 Series Datasheet.

9 PACKAGE DIMENSIONS

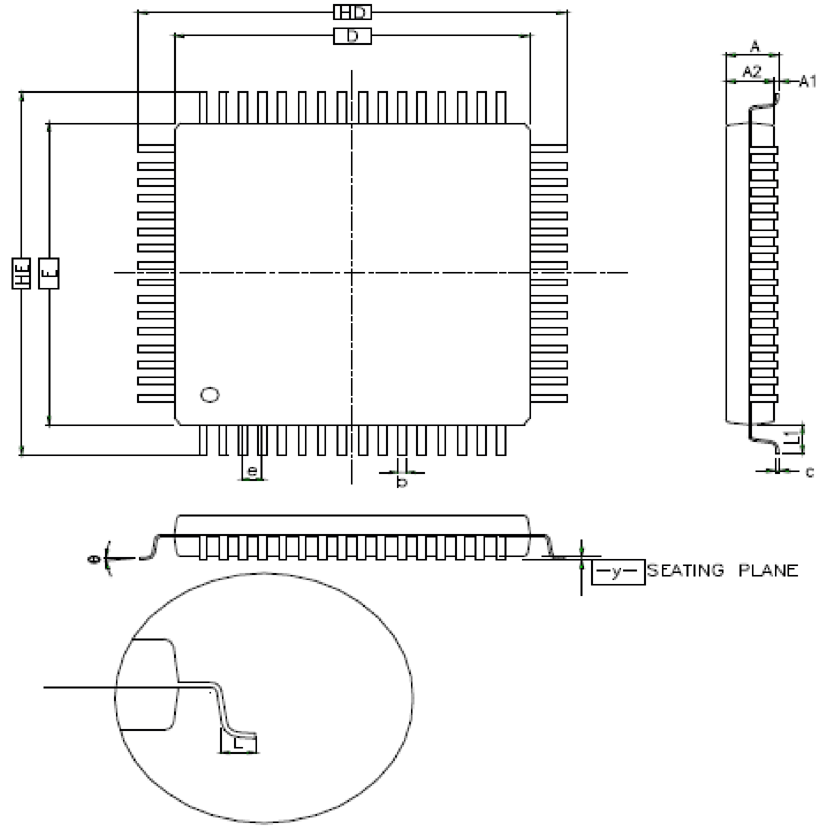
9.1 100L LQFP (14x14x1.4 mm footprint 2.0 mm)



Controlling Dimension : Millimeters

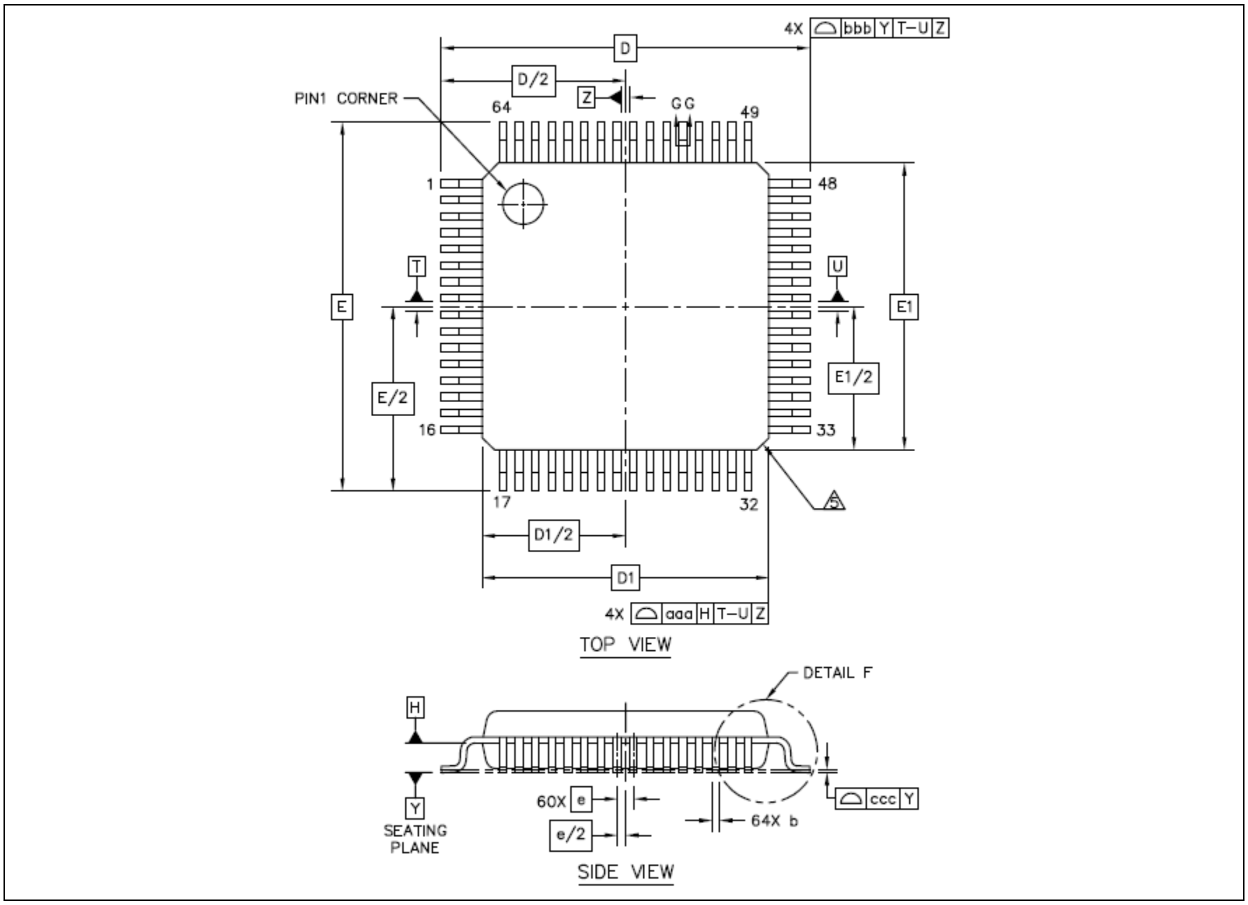
Symbol	Dimension in inch			Dimension in mm		
	Min	Nom	Max	Min	Nom	Max
A	—	—	0.063	—	—	1.60
A1	0.002	—	—	0.05	—	—
A	0.053	0.055	0.057	1.35	1.40	1.45
b	0.007	0.009	0.011	0.17	0.22	0.27
c	0.004	0.006	0.008	0.10	0.15	0.20
D	0.547	0.551	0.556	13.90	14.00	14.10
E	0.547	0.551	0.556	13.90	14.00	14.10
e	—	0.020	—	—	0.50	—
H_D	0.622	0.630	0.638	15.80	16.00	16.20
H_E	0.622	0.630	0.638	15.80	16.00	16.20
L	0.018	0.024	0.030	0.45	0.60	0.75
L1	—	0.039	—	—	1.00	—
y	—	—	0.004	—	—	0.10
θ	0°	—	7°	0°	—	7°

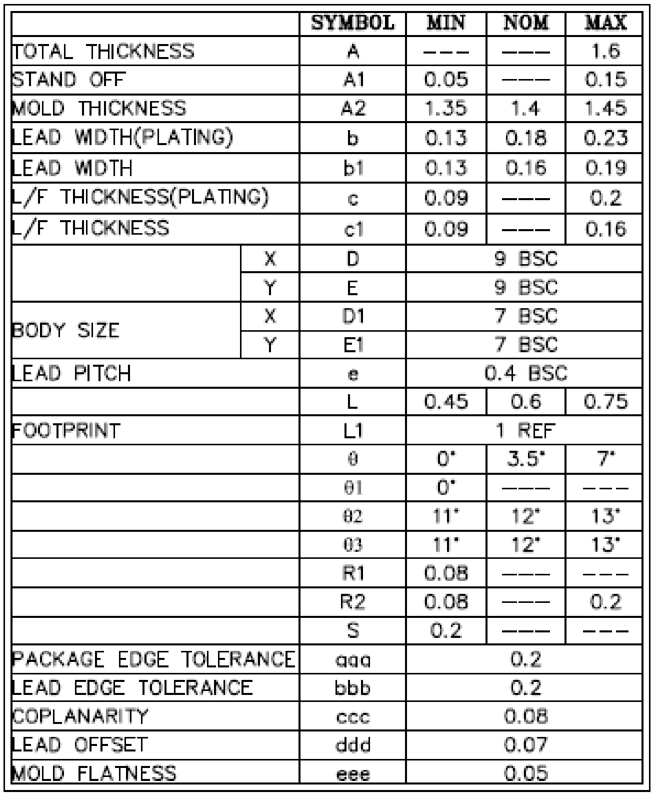
9.2 64R LQFP(10x10x1.4 mm footprint 2.0 mm)



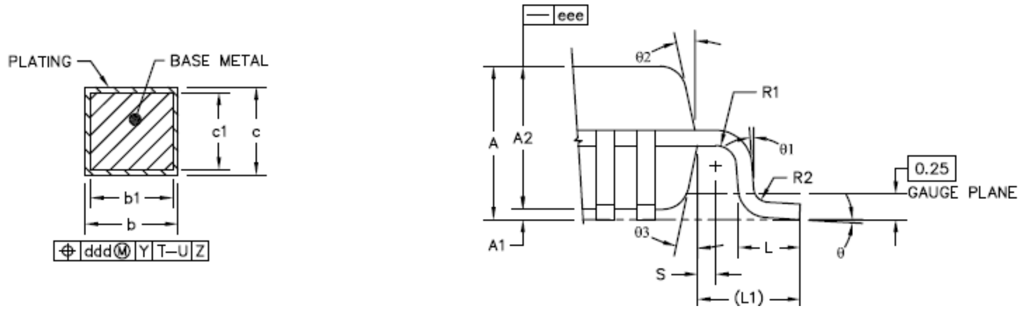
Symbol	Dimension in inch			Dimension in mm		
	Min	Nom	Max	Min	Nom	Max
A	—	—	0.063	—	—	1.60
A_1	0.002	—	0.006	0.05	—	0.15
A_2	0.053	0.055	0.057	1.35	1.40	1.45
b	0.007	0.008	0.011	0.17	0.20	0.27
c	0.004	—	0.008	0.09	—	0.20
D	—	0.393	—	—	10.00	—
E	—	0.393	—	—	10.00	—
$\boxed{e}$	—	0.020	—	—	0.50	—
H_D	—	0.472	—	—	12.00	—
H_E	—	0.472	—	—	12.00	—
L	0.018	0.024	0.030	0.45	0.60	0.75
L_1	—	0.039	—	—	1.00	—
y	—	0.004	—	—	0.10	—
θ	0°	3.5°	7°	0°	3.5°	7°

9.3 64S LQFP (7x7x1.4 mm footprint 2.0 mm)

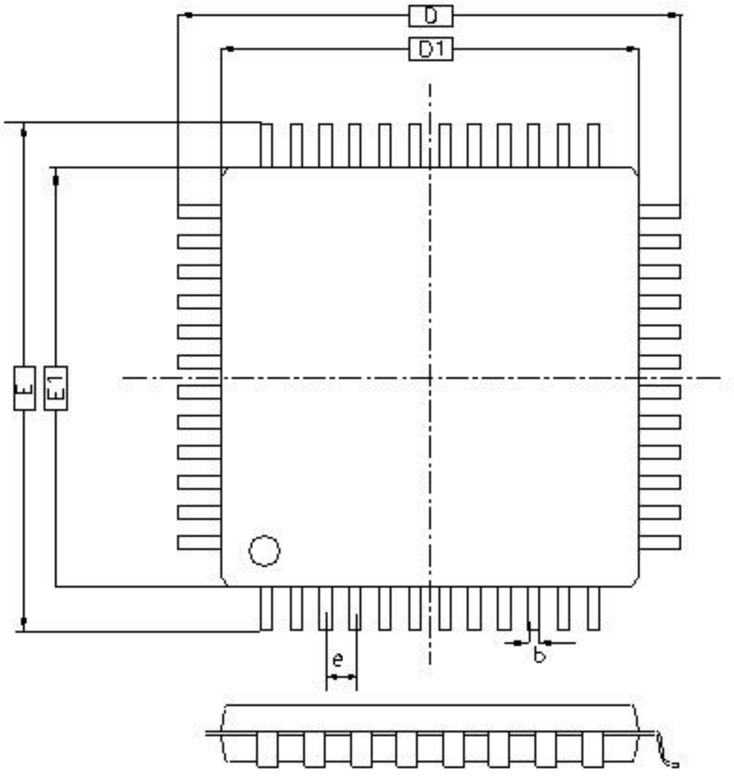


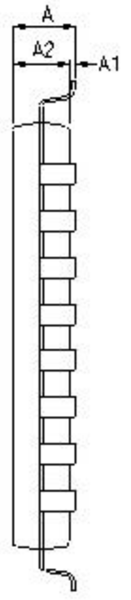


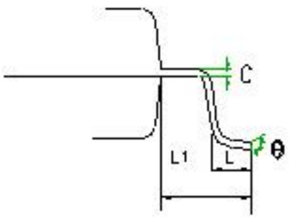
		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	---	---	1.6
STAND OFF		A1	0.05	---	0.15
MOLD THICKNESS		A2	1.35	1.4	1.45
LEAD WIDTH(PLATING)		b	0.13	0.18	0.23
LEAD WIDTH		b1	0.13	0.16	0.19
L/F THICKNESS(PLATING)		c	0.09	---	0.2
L/F THICKNESS		c1	0.09	---	0.16
	X	D	9 BSC		
	Y	E	9 BSC		
BODY SIZE	X	D1	7 BSC		
	Y	E1	7 BSC		
LEAD PITCH		e	0.4 BSC		
		L	0.45	0.6	0.75
FOOTPRINT		L1	1 REF		
		θ	0°	3.5°	7°
		θ1	0°	---	---
		θ2	11°	12°	13°
		θ3	11°	12°	13°
		R1	0.08	---	---
		R2	0.08	---	0.2
		S	0.2	---	---
PACKAGE EDGE TOLERANCE		aaa	0.2		
LEAD EDGE TOLERANCE		bbb	0.2		
COPLANARITY		ccc	0.08		
LEAD OFFSET		ddd	0.07		
MOLD FLATNESS		eee	0.05		



9.4 48L LQFP (7x7x1.4 mm footprint 2.0 mm)



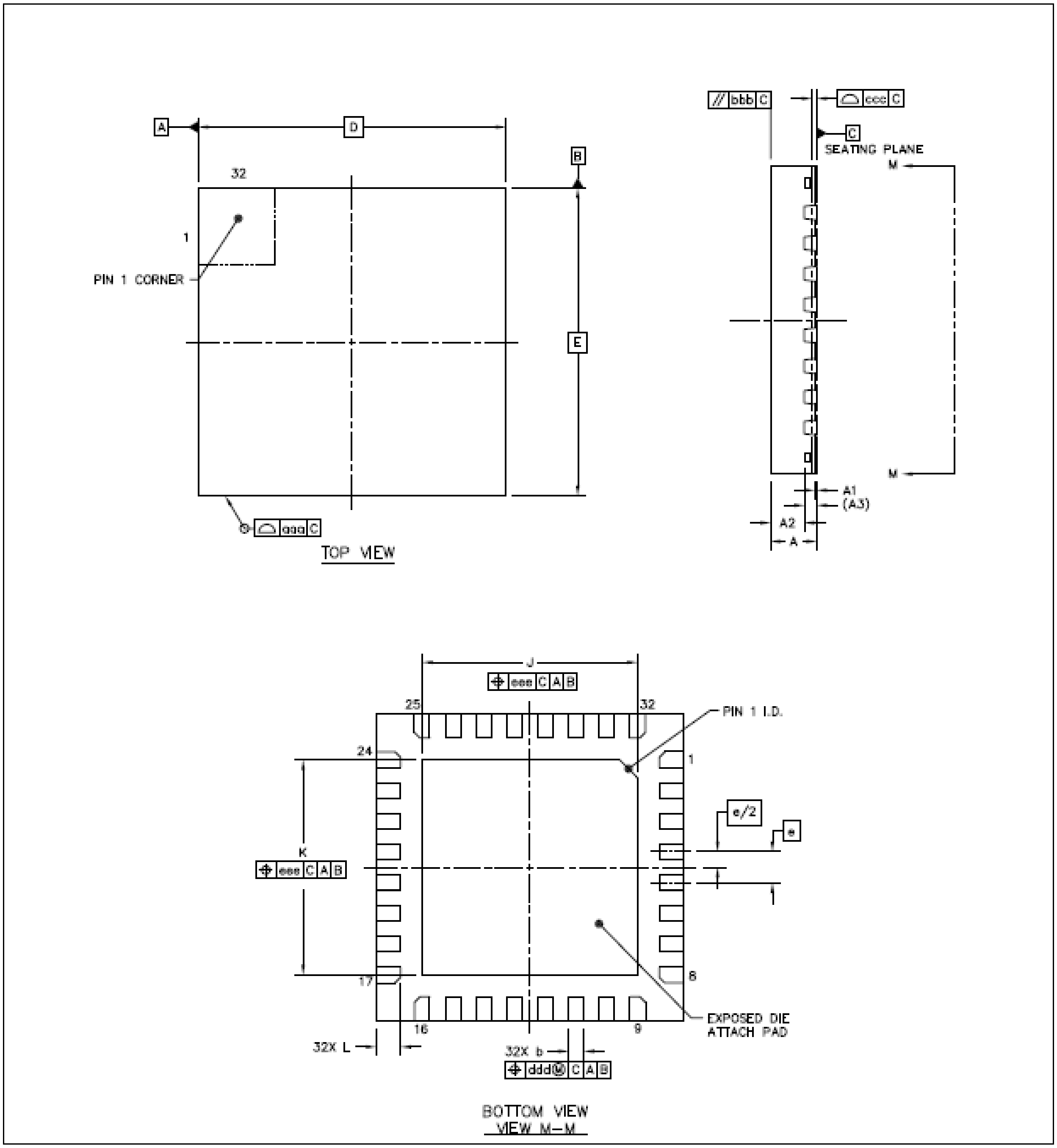


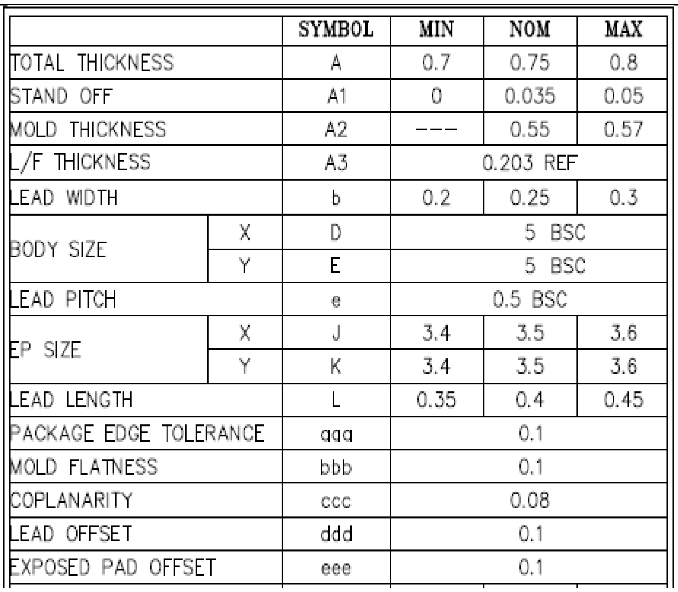


COTROL DIMENSIONS ARE IN MILLIMETERS.

SYMBOL	MILLIMETER			INCH		
	MIN	NOM.	MAX.	MIN	NOM.	MAX.
A	—	—	1.60	—	—	0.063
A1	0.05	0.10	0.15	0.002	0.004	0.006
A2	1 35	1.40	1.45	0.053	0 055	0.057
D1	6.90	7.00	7.10	0.272	0.276	0.260
E1	6 90	7.00	7 10	0.272	0.276	0.260
e	0.35	0.50	0.65	0.014	0.020	0.260
D	8.9	9.00	9.10	0.350	0.354	0.358
E	8.9	9.00	9.10	0.350	0.354	0.358
L	0 45	0.60	0 75	0.018	0 024	0 030
L1	—	1 00	—	—	0.039	—
C	0.09	—	0.20	0.0035	—	0.0079
θ	0°	—	7°	0°	—	7°
b	0.17	0.22	0.27	0.007	0.0087	0.011

9.5 33L QFN (5x5x1.4 mm footprint 2.0 mm)





		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.7	0.75	0.8
STAND OFF		A1	0	0.035	0.05
MOLD THICKNESS		A2	---	0.55	0.57
L/F THICKNESS		A3	0.203 REF		
LEAD WIDTH		b	0.2	0.25	0.3
BODY SIZE	X	D	5 BSC		
	Y	E	5 BSC		
LEAD PITCH		e	0.5 BSC		
EP SIZE	X	J	3.4	3.5	3.6
	Y	K	3.4	3.5	3.6
LEAD LENGTH		L	0.35	0.4	0.45
PACKAGE EDGE TOLERANCE		aaa	0.1		
MOLD FLATNESS		bbb	0.1		
COPLANARITY		ccc	0.08		
LEAD OFFSET		ddd	0.1		
EXPOSED PAD OFFSET		eee	0.1		

10 REVISION HISTORY

Date	Revision	Description
2014.03.28	1.00	1. Initial version
2014.05.08	1.01	1. Modified some typos and format
2014.09.02	1.02	1. Modified the pin description for LCD_Vx in section 4.4 2. Modified the CHXT_GAIN(Config0[15:13]) bit description in section 6.6.4.7 3. Modified all PWM1 group to PWM0 group in section 6.10 4. Modified “PWM1 channel 2 and 3” to “PWM0 channel 2 and 3” in section 6.10 5. Updated Table 6-14 Watchdog Time-out Interval Selection in section 6.11.4 6. Updated I^{2}C pin configuration in section 6.16.3 7. Removed redundant description of I^{2}C protocol in section 6.16.4.1 8. Modified the I2C_STS(I2CON[4]) bit description in section 6.16.6 9. Modified External Capacitor ladder type application circuit in section 6.18.7 10. Modified some typos and format
2015.01.15	1.03	1. Updated ADC channel number in NANO102 feature list in Chapter 2 2. Corrected typo in NANO102 64-pin sequence in section 4.4 3. Updated PE_L_MFP bit field number in section 6.4.5 4. Updated the bit 31 description in Config0 to “Should be set to 1” in section 6.6.4.7 5. Updated all power related pins from “VDD, VSS, AVDD, AVSS, VTEMP and VLCD” to “V_{DD}, V_{SS}, AV_{DD}, AV_{SS}, V_{TEMP} and V_{LCD}” in the TRM
2015.05.19	1.04	1. Added a note in all clock source block diagram of all peripherals sections that “Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.” 2. Added Systick related description and register in section 6.1 3. Updated power distribution diagram in section 6.4.3.3 4. Corrected Px_MFP bit field length in section 6.4.5 5. Updated LCD application circuit in section 6.18.7 6. Updated bit field description and note for PWDMOD(ADCPWD[3:2]) in section 6.19.6 7. Updated bit field description of register PWM_PDMACH0 and PWM_PDMACH2 in section 6.10.7 8. Changed chapter order for Chapter 8 and 9 9. Corrected typos and updated document format
2016.03.31	1.05	1. Modified the TEMPCTL register address offset value in section 6.4.4 and 6.4.5 2. Added Internal Reference Voltage description in section 6.19.4.15
2020.04.17	1.06	1. Added application circuit in chapter 7 2. Added notes about the hardware reference design for ICE_DAT, ICE_CLK and nRESET pins in section 4.4 and chapter 7

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, “Insecure Usage”.

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer’s risk, and in the event that third parties lay claims to Nuvoton as a result of customer’s Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.